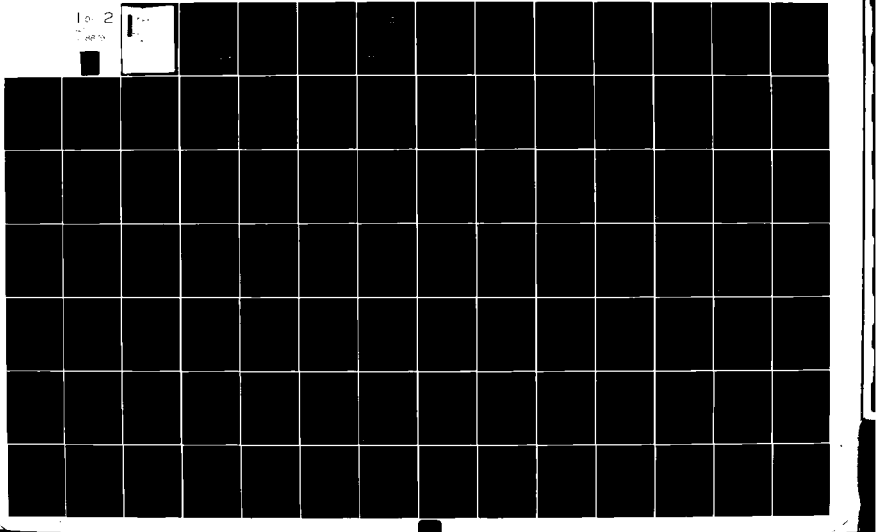
AD-A118 879 ALPHA OMEGA GROUP INC SILVER SPRING MD F/G 5/2
DEVELOPMENT OF A USER-ORIENTED DATA CLASSIFICATION FOR INFORMAT--ETC(U)
JUN 82 N00014-82-C-0129
UNCLASSIFIED AOG82-ONR-1 NL



Alpha Omega Group, Inc.
8121 Georgia Avenue, Suite 406
Silver Spring, Maryland 20910

Final Report

Prepared for:

Materials Technology Project
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

DEVELOPMENT OF A USER-ORIENTED DATA CLASSIFICATION FOR INFORMATION SYSTEM DESIGN METHODOLOGY

30 June 1982

AOG-ONR-1

DEVELOPMENT OF A USER-ORIENTED DATA CLASSIFICATION FOR INFORMATION SYSTEM DESIGN METHODOLOGY

Alpha Omega Group, Inc.
8121 Georgia Avenue, Suite 406
Silver Spring, Maryland 20910

30 June 1982

Final Report for Period 1 January 1982 - 30 June 1982

Distribution:

Procuring Contracting Officer, Office of Naval Research
Materials Technology Project Manager, Office of Naval Research
Defense Technical Information Center
Naval Research Laboratory
Office of Naval Research, Eastern/Central Regional Office

Prepared for:

Materials Technology Project
Office of Naval Research
800 North Quincy Street
Arlington, VA 22217

Monitored by:

DCASMA Baltimore
300 East Joppa Road
Towson, MD 21204

REPRODUCTION LIMITATIONS

Reproduction and distribution of this Report is unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AOG82-ONR-1	2. GOVT ACCESSION NO. AD-A118 879	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DEVELOPMENT OF A USER-ORIENTED DATA CLASSIFICATION FOR INFORMATION SYSTEM DESIGN METHODOLOGY		5. TYPE OF REPORT & PERIOD COVERED Final Report 4 Jan 1982 - 30 Jun 1982 6. PERFORMING ORG. REPORT NUMBER AOG82-ONR-1
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s) N00014-82-C-0129
9. PERFORMING ORGANIZATION NAME AND ADDRESS Alpha Omega Group, Inc. 8121 Georgia Avenue, Suite 406 Silver Spring, Maryland 20910		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Materials Technology Project, Office of Naval Research, 800 N. Quincy St. Arlington, Virginia 22217		12. REPORT DATE 30 Jun 1982 13. NUMBER OF PAGES 124
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) DCASMA Baltimore 300 East Joppa Road Towson, Maryland 21204		15. SECURITY CLASS. (of this report) NA 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Procuring Contracting Officer, Office of Naval Research; Materials Technology Project Manager, Office of Naval Research; Defense Technical Information Center; Naval Research Laboratory; Office of Naval Research, Eastern/Central Regional Office		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) CLASSIFICATION — DATABASE DESIGN FACETED CLASSIFICATION — DATA DICTIONARIES INFORMATION SYSTEM DESIGN — USER-ORIENTED SYSTEMS REQUIREMENTS ANALYSIS — DATA CLASSIFICATION		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) A comprehensive review of information system design methodologies demonstrates that there is a need for new and improved approaches particularly in the early stages of system design. Specifically, a methodology is needed that will collect and organize data from and about the information environment in order to present a coherent, systematic, and dynamic picture of an enterprise and		

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Figure 6. Report Documentation Page.

its activities that will support the development of requirements statements, and later, database and data dictionary design and/or use.

A new methodology is described, drawing on the principles and structures of faceted classification as used primarily with complex bibliographic databases. Faceted classification offers a rigorous analytical procedure by which complex records or data statements can be broken down into basic data elements. These elements are then clustered on the basis of common characteristics into single-characteristic facets of the universe of discourse, to produce a multi-faceted classification of the components of records in the information environment.

Highly complex data element or record descriptions can thus be assembled at need from these basic components, without recourse to a long and cumbersome classification of preconstructed statements. Because facets can be developed individually, as simple, mutually exclusive hierarchies, a faceted classification is easy to construct and maintain. Changes in the information environment can be reflected by restructuring only the facets affected, without causing changes in other facets. Similarly, the same changes can be reflected in the set of complex data element descriptions supported by the classification, since each description is articulated explicitly to show components in separate facets.

A computer-aided instructional sequence is described that can guide an analyst in building a classification for his information environment, and in applying it to complex data elements, to build a coherent and dynamic picture of the environment.

SUMMARY

A comprehensive review of information system design methodologies demonstrates that there is a need for new and improved approaches, particularly in the early stages of system design. Specifically, a methodology is needed that will collect and organize data from and about the information environment in order to present a coherent, systematic, and dynamic picture of an enterprise and its activities that will support the development of requirements statements, and later, database and data dictionary design and/or use.

A new methodology is described, drawing on the principles and structures of faceted classification as used primarily with complex bibliographic databases. Faceted classification offers a rigorous analytical procedure by which complex records or data statements can be broken down into basic data elements. These elements are then clustered on the basis of common characteristics into single-characteristic facets of the universe of discourse, to produce a multi-faceted classification of the components of records in the information environment.

Highly complex data element or record descriptions can thus be assembled at need from these basic components, without recourse to a long and cumbersome classification of preconstructed statements. Because facets can be developed individually, as simple, mutually exclusive hierarchies, a faceted classification is easy to construct and maintain. Changes in the information environment can be reflected by restructuring only the facets affected, without causing changes in other facets. Similarly, the same changes can be reflected in the set of complex data element descriptions supported by the classification, since each description is articulated explicitly to show components in separate facets.

A computer-aided instructional sequence is described that can guide an analyst in building a classification for his information environment, and in applying it to complex data elements, to build a coherent and dynamic picture of the environment.

Sections 9, 10, and 11 of this Report describe the methodology in detail.

CONTENTS

		Page
	SUMMARY	1
SECTION 1:	Identification and Significance of the Problem and the Proposed Solution	3
SECTION 2:	Technical Work	6
SECTION 3:	Brief Review of Database Design	8
SECTION 4:	A Review of Requirements Analysis Methodologies	24
SECTION 5:	Summary of the Review of Requirements Analysis Methodologies	44
SECTION 6:	Classification Theory and Facet Analysis	49
SECTION 7:	The Need for Faceted Classification in the Information System Environment	60
SECTION 8:	An Analytico-Synthetic Model of the Information Environment	64
SECTION 9:	Development of the Facet Structure and the Classification Scheme	72
SECTION 10:	Application of the Classification to the Information Environment Data	86
SECTION 11:	Development of User Friendly Aids in Classification Development	89
SECTION 12:	Implementation of the System and the Use of the New Methodology	96
	REFERENCES	98
	APPENDIX A: Development of a Data Element Dictionary and Locator System	
	APPENDIX B: Glossary	

SECTION 1

IDENTIFICATION AND SIGNIFICANCE OF THE PROBLEM AND THE PROPOSED SOLUTION

The high failure rate of large scale data systems and the resulting user dissatisfaction has become a major concern in recent years. The problems can be traced to the following aspects of modern data system development:

- o difficulties in integrating the data throughout the enterprise;
- o poor data structuring once data have been identified and some usages understood;
- o difficulties in identifying stored data.

These are really manifestations of the same basic problem - lack of understanding of the data semantics in the organization. The problem of dealing with the complexity of data structures and their implicit relationships is as fundamental to modern database and information systems design as the problems of program component relationships of twenty years ago.

There have been many attempts in the past few years to define the requirements of the information system and then to use this definition of the essential processes as a means of defining the system, but these (though they have found active user proponents) have been neither highly successful nor widely accepted.

It is our belief that the major reasons for the lack of success of these attempts are:

- o the concentration on process that tends to ignore the data, and that leads to the dual problem of misunderstanding among system developers and a lack of use of data across interfaces of the organization;
- o the analysis of systems generally stops at the end of the requirements gathering phase. The results, though voluminous, are seldom used in the development of the databases or the ultimate system.

Many modern information systems use a dictionary system to record the "meta data" (i.e., data about data), including the name and natural language definition of the element, and the requirements analysis sometimes feeds the dictionary with these "facts". However, the software configuration management information (i.e., the information about the requirements and how they relate to the software and its testing, with a careful trace of all changes to the requirements and software) seldom uses the dictionary or is even absent from the system in an automated sense.

We believe the use of modern information science techniques developed for the classification of complex elements, in conjunction with more modern data systems techniques, will lead to better understanding, discovery, use, and control of data. This, coupled with the ability to structure the data for ease of retrieval and update, will result in the ultimate development of better user systems.

Our hypothesis that there were gaps in the range and depth of the various methodologies available to support information system design was borne out by the investigation. These gaps were especially noticeable in the early stages of the design process. No comprehensive methodology was found that could collect information about an enterprise, and specifically about its information environment, and then organize it in such a way as to support a clear and coherent picture of the environment.

If such an exercise were undertaken in the present state-of-the-art, it would have to be undertaken on an entirely empirical basis. Using existing methods, it is unlikely that any attempt could be made to put together a coherent picture of the information environment of an enterprise until after all the information had been collected, and situations exist in which it is impossible to determine when all the information has been collected. Further, it is also unlikely that the picture so put together would be complex enough, that future changes in the enterprise and its information environment could be assimilated easily or quickly.

Consequently, the Project Team proposed to develop a new way of putting together a picture of the information environment of an enterprise, by drawing on methodologies from library and information science, already available and tested for handling large and complex databases. The proposed methodology, which is described in full in the later sections of this Report, has three components:

- a classification scheme that contains basic data elements (BDE) organized in clusters or facets, each of which displays a single characteristic. For example, DOORS is a BDE in the facet BUILDING COMPONENTS, along with WINDOWS, CEILING TILES, etc.; WOOD is a BDE in the facet BUILDING MATERIALS, along with PLASTIC, STONE, etc.; FINISHING is a BDE in the facet of BUILDING OPERATIONS, along with ASSEMBLING, INSTALLING, etc.; and NUMBER is a BDE in the facet of MEASUREMENT UNITS, along with TONS, FEET, GALLONS, etc.

- a set of complex data elements (CDE), each of which describes an event or phenomenon in the information environment, formed by assembling BDE's from the classification scheme (e.g., NUMBER OF WOODEN DOORS FINISHED; NUMBER OF PLASTIC CEILING TILES INSTALLED, etc.); the total set of CDE's would describe the total information environment of the enterprise.

- an instructional sequence (preferably computer-aided) to support the development of the classification, and its application to the information environment.

Because the classification is what is called analytico-synthetic, or more specifically, faceted, the general structure of the classification can be determined from only a preliminary sample of information from the environment. It is thus predictive, to facilitate and accelerate both its own development, and the development of the complex data element descriptions, and can accommodate changes or additions at any time without disruption of the general structure, because changes are made within the independent mini-hierarchies of individual facets.

The practical result for the enterprise of the development of the faceted classification and the construction of CDE descriptions is a coherent picture of all events and phenomena in its information environment. The classification reveals errors or confusions in basic data element names: homonyms, synonyms, clerical errors in names and the transcription of names. The set of complex data element descriptions reveals redundancies and/or gaps in collecting, handling, and reporting information about the enterprise.

If the facts about the entire information system design and changes could be stored in a configuration management database, then the active dictionary could be used for change control, and better control would be possible over the entire software development life cycle.

Finally, if dictionaries, database management systems, and configuration management could be implemented as a single unit through the use of classification, the resulting tool would be a very substantial aid in software production and its control.

SECTION 2

TECHNICAL WORK

The research began with a literature review of four areas:

- o database design methodology;
- o requirements analysis methodology;
- o faceted classification theory;
- o the development and application of faceted classification in appropriate and similar areas.

The Project Team then examined the information system design process, in order to determine:

- o how data from an information environment were collected and organized;
- o how these data then presented a picture of the information environment;
- o what was the shortfall in the methodologies now in use;
- o what characteristics did these methodologies share, and did they need new features.

The Project Team made a genuine effort to find a working system, or even a conceptual basis for a system in the developmental stage, that could handle complex data at the requirements analysis stage satisfactorily. As the following pages show, no such system was found.

Sections 3-5 of this Report present the results of the research in the database and requirements analysis methodologies areas.

Section 6 presents the results of the research on existing classification theory, principles, and application in appropriate areas.

Research then continued in the application of faceted classification principles, of facet analysis techniques, and of faceted classification structures to the collection and organization of data about an enterprise and its information environment. The first part of this research was a feasibility study to see if the approach were appropriate. Faceted classification as a descriptive language was compared to previous methods of describing data in the information environment. Sections 7-8 present the Project Team's judgment on the appropriateness of faceted classification structure as at least a working hypothesis for a solution to the problem in hand.

The Project Team then considered how a faceted data classification could and would be constructed. This involved a thorough investigation and adaptation of methods previously used to construct faceted classification schemes in bibliographic environments. It was necessary to examine carefully each stage of construction because of the difference in environments, and because in the present case the end use would be at least as a front end to a machine sensible environment.

The Project Team also examined the other side of the construction of a faceted classification: its application to the data elements in the information environment.

Section 9 is a full discussion of the Project Team's deliberation on the way in which faceted classification works and could be made to work in the development of the structure and detail of a data classification.

Section 10 is an outline account (because the method differs slightly with each information environment) of the application of the classification to the information environment.

Finally the Project Team considered the ways in which an automated user friendly system could be used to aid in the development of a data classification. The design and development of such a system was not part of the Project, but the Project Team felt that the Report should include not only a detailed account of a method of using faceted classification to organize data and complex data element descriptions of the information environment; it should also include some indication of what practical implementation of the method could offer. Section 11 describes a way (to be developed and tested) in which the method described in Sections 9-10 would be automated and presented to the user in a user-friendly mode.

Section 12 offers suggestions for demonstrating and testing the methodology.

SECTION 3

BRIEF REVIEW OF DATABASE DESIGN

3.1 INTRODUCTION

Before examining the relevance and utility of a new kind of data classification for information system design, it is necessary to examine existing methodologies: at the broad level of information system and database design; and at the specific level of those methodologies that address only the early stages of such a design.

The evolution of much of information system design methodologies has been *ad hoc* at best. Conventional database design has focused on the integration and efficient management of data collections that have already been well-defined in the enterprise and formerly were processed manually or by a file management system. These manual methods (involving masses of data contained in hand- or typewritten records stored in file drawers) were then automated with little change in the structure of the records and files.

The ANSI/X3/SPARC report [ANS75] defines a database conceptual schema as a "long term, unrestricted model (or view) of the enterprise", implying that a database design involves a conceptual modelling of an enterprise. However, current methods of database design are not satisfactory for modelling complex and not-explicitly-defined phenomena that can be found in new, expanding, and diversifying applications. With the exception of work by Clemons [CLE77,79], surprisingly little has been done in the area of conceptual schema design in supporting the earlier stages of the design, when phenomena in the universe of discourse must be analyzed and requirements for the database system specified.

In the current state-of-the-art, the methods used in the design of database applications are supported neither by a scientific foundation nor by an engineering discipline [YAO78]. However, several sophisticated data models have been proposed. A data model is a set of concepts and constructs by which the contents of a database and relationships within it can be described and manipulated: data structures, operations, and constraints. The data model supports the design of a database schema, by supporting definition that in turn supports actual database implementation. The database schema describes what kinds of data are contained in the database and the constraints and rules to which the data must conform. Database design is focused on the database schema design and may be divided into the three distinct stages corresponding to the three levels in the ANSI/SPARC DBMS diagram as shown in Fig. 3:1.

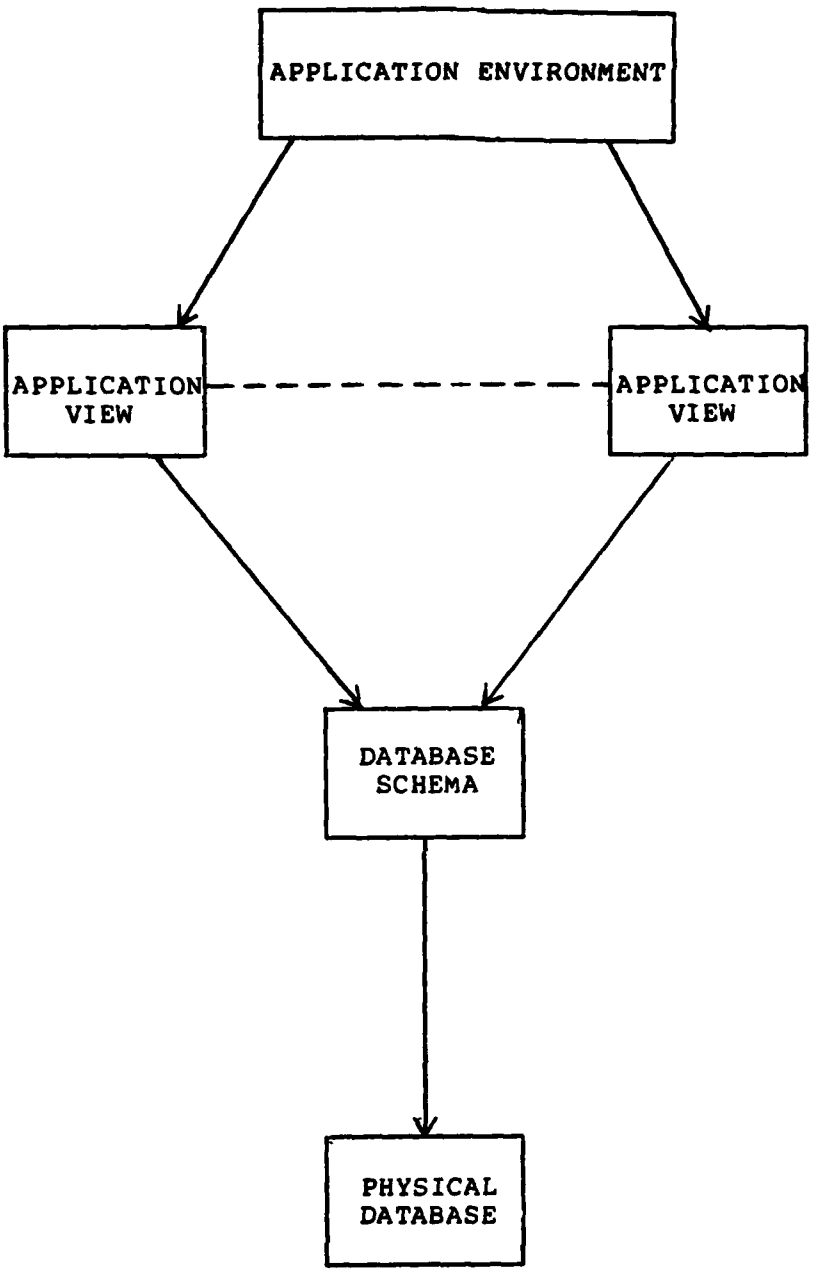


Figure 3:1 Database Design Stages

One purpose of the requirements analysis stage is to analyze the real world problem and its environment. To achieve this, the necessary components of the database are first identified. Data and processing needs of all potential database users are then analyzed in order to describe the necessary database components (termed application views). In fact, many of the requirements analysis methodologies described in Section 4 focus on the translation of already discerned requirements into a schema and subsequent physical design, rather than on discerning the requirements themselves.

The database schema design stage involves the integration of all the concepts that are necessary to support the various application views. At this stage, it is important to define how concepts are related to one another without considering unnecessary implementation detail.

The physical database design is the mapping of the schema model on to physical computing devices generally considering the efficiency and timing needs. In this effort we are not concerned with the physical design level.

3.2 DATABASE DESIGN METHODOLOGY

Database design methodology has been defined as the systematic process by which one level of data abstraction can be mapped to the next [BUC79]. The complexity of database design is strongly influenced by the number of types of objects to be stored, the number of relationships among them, and the kinds of processing required.

Present database design methodologies have several distinct characteristics and origins; they may be characterized by the following unordered and tentative list of properties:

- design strategy - analysis;
- requirements specification procedures;
- aggregation process and abstraction mechanisms;
- separation of the information and processing requirements;
- scope of the model;
- number of abstraction levels;
- use of functional dependencies;
- use of roles;
- treatment of security, update propagation, etc.;

- o modularity;
- o completeness;
- o adaptability to automated design tools.

From our point of view, the first four properties listed above are the most important. These will now be discussed in more detail.

3.2.1 Design Strategy

Design strategy is concerned with the choice of either the top-down or bottom-up approach. Database technology conventionally uses a bottom-up strategy. However, for engineering purposes the top-down and requirements-first approaches are generally better. The top-down approach is based on a preconceived mental model of reality. As this Report suggests elsewhere, the use of facet analysis supports a combination of top-down and bottom-up in predefined sequence that is likely to offer the most practical strategy.

3.2.2 Requirements Specification Procedure

For a requirements-first database design methodology, it is necessary to have a good requirements gathering and analyzing procedure. This need is widely recognized but (as noted in Section 4) is usually not supported. Frequently, emphasis is on the development of requirements specification languages and automated analysis procedures, which is helpful only when the physical reality has been correctly perceived. The problems which are most difficult to eliminate arise from false perceptions of reality and the lack of appropriate tools for communication between the 'database naive' user and the 'application naive' database designer.

3.2.3 Aggregation Process and Abstraction Mechanisms

A major difference among existing methodologies is their approach to the aggregation process. Two basically different approaches are:

- o to design a schema first and then derive views from it;
- o to model the views of different users first and then to integrate them into a schema.

Some researchers argue that for large engineering applications the second approach appears to be better [BUC79].

Fundamentally, in the requirements analysis and conceptual design

stages we need abstraction mechanisms for developing complex systems. An abstraction of some phenomena is a general name given to a collection of details that can be conveniently named as a whole. For example, PERSON is the abstraction of a collection of people; RUN represents a collection of movements of an animal's body.

3.2.4 Separation of the Information and Processing Requirements

Database design is concerned almost entirely with the definition of abstract objects, rather than of abstract operations. Abstract objects are defined first, and it is assumed that it is relatively easy to cluster abstract operations around the objects to which they apply. Therefore, the design of abstract operations is usually beyond the scope of database design methodologies.

There appears to be a consensus among database design experts that information perspective should be given priority over processing requirements. Using the resulting database schema, the processing requirements are developed by defining a processing-oriented schema (the view) interfaced with an information-oriented schema.

3.3 CURRENT DESIGN METHODOLOGIES

We shall now briefly review some recently developed database design methodologies as described by the Smiths [SMI78], Kahn [KAH78], and Bubenko [BUB77].

3.3.1 The Smiths' Methodology

The methodology proposed by the Smiths [SMI78] ignores the requirements gathering and analysis stage. It assumes that the concepts to be integrated are available and are fully understood by the database designer for whom data abstractions are the fundamental means of designing a database system.

The mechanisms for constructing an abstract data object are:

- generalization of the common properties, ignoring differences in a class of objects;
- aggregation, by naming a relationship among some objects.

Repeated application of generalization and aggregation results in abstraction hierarchies which can be visualized as lying in perpendicular intersecting planes. Abstraction is applied to the names provided by the requirements stage; these abstractions carry a large part of the semantics. At the conceptual level an object does not have a fixed interpretation; it can be viewed as

an entity, a component, a relationship, a category, an attribute, or an instance, depending solely on the viewpoint of the user. The activities of the abstraction steps are highly intuitive and depend on the insight and abstraction capabilities of the database designer. Basic steps of the methodology are:

generalization
aggregation
instance identification
instance expression in terms of an abstract syntax

3.3.2 Kahn's Methodology

In Kahn's structural logical database design methodology [KAH78], the overall design problem is separated at an early stage into two parallel perspectives:

- the information structure, describing interconnections within an organization;
- the usage perspective, concerned with satisfying the processing requirements within an organization.

The process is considered to consist of six levels, with five steps to progress from one level to the next:

LEVELS	STEPS
Real world requirements	
	Requirements step
Local information structures	
	Entity and Relationship step
Global information structures	
	Entity structure step
Entity structure	
	Refinement step
Revised entity structure	
	DBMS accommodation step
Logical database structure	

The Requirements step consists of the following activities:

- o a gross system analysis;
- o selection of a design path between an information structure and processing requirements analysis;
- o requirements technique selection;
- o requirements collection and specification;
- o requirements analysis to produce an information requirements document. (The developers have not worked out the details of this activity.)

The Entity and Relationship step starts by designing global entities which represent aggregated needs of the organization. However, this difficult aggregation task (forming a nonredundant collection of entities) is left largely to the designer's intuition. An equivalent step is then performed to create global relationships and then eliminate all redundant relationships.

In the Entity Structure step, the global/conceptual information structure is mapped into functional dependencies, and from there into normalized relations.

The Refinement step introduces controlled redundancy, ensures adequate processing, and accommodates security and volatility of the data.

The DBMS Accommodation step comprises all those actions necessary to express the database schema in the data dictionary language of a chosen DBMS.

3.3.3 Bubenko's Methodology

The Inferential Abstract Modelling (IAM) approach [BUB75] can be divided into seven iteratively executable groups of activities:

- o collection and specification of information requirements;
- o entity classification;
- o specification of functional dependencies;
- o abstract object specification, integration, and analysis;
- o implied information analysis;

- o derivability (precedence) analysis;
- o transformation to an external name-based model.

The first step is mainly a narrative gathering of information requirements, with their formulation in terms of queries and transactions. The entities are next classified into concept classes by extracting grammatically understood subjects and objects of the query and transaction descriptions. Such grouping into concept classes is highly intuitive and difficult to formalize because of the application dependence of this step.

The various kinds of functional dependencies are recognized by the IAM methodology, helping, in this way, to identify and specify them.

The abstract objects are then determined from the previous steps. Iteration is often needed at this point. The information analysis leads to the identification of implied and derived associations and their corresponding rules. In the last step, the abstract model is mapped into a name-based model through the introduction of name sets.

3.4 NEW TRENDS IN DATABASE DESIGN

The primary direction of new trends in database systems design is to aid currently rather intuitive, early design stages of requirements analysis and conceptual modelling. Thus, high-level data models have been recently proposed that should or could enable designers to capture more semantics in data structures [HAM78, COD79, ROU79]. Also, high-level semantics-oriented modelling methods are being developed in order to support identification of not well-defined structures in problem domains [WIL79].

In this section, we shall briefly review two semantics-oriented efforts, those of Wilson and Roussopoulos respectively, that are going in the same general direction as our approach of interfacing database design with requirements analysis.

3.4.1 Wilson's Semantics-based Requirements and Design Method

Wilson's methodology [WIL79] consists of 30 very specific steps organized into six phases, (1, 2, and 3 covering requirements analysis and 4, 5, and 6 system design):

1.	Subject Analysis	)
		)
2.	Data Analysis	) Requirements Analysis
		)
3.	Process Analysis	)
4.	Data Structuring	)
		)
5.	Process Structuring	) System Design
		)
6.	System Packaging	)

The key feature of this method is the semantic analysis of the original problem. This is understood as the activity of creating formal definitions of the terms, words, and other symbols used to discuss the requirements for a data processing system. Wilson's semantic model is focused on the recognition of:

- o certain categories of concepts;
- o certain key relationships among concepts;
- o certain rules of inference based on these concepts and their relationship characteristics.

Five categories of concepts are delineated: entity, event, relationship, attribute, and value. There are five categories of relationships among concepts: attribute of entity/event, attribute of relationship, value of attribute, from-subject and to-subject relationships.

Semantic analysis of the terms used to discuss the subject area is performed during the subject analysis phase. Problem-specific terms are first listed and then classified into basic concepts; next, relationships among the concepts are identified. Data and process analysis are separated and carried out in several detailed steps.

3.4.2 Roussopoulos' Conceptual Schema Definition Language (CSDL) and Semantic Networks

The purpose of Roussopoulos' CSDL [ROU79] is to simplify the translation of unstructured knowledge about the data and its uses into a formal schema, in the conceptual modelling stage of database systems development. In this stage, analysts and designers are concerned with the behavior of the entities of the application and the relationships among them, but not by their physical representation in the computer system.

CSDL provides a unified notation for both data and process description. It can be expressed in three different ways, but probably the graph notation of the semantic networks is easiest

to follow. A semantic network is a graph consisting of labelled nodes, edges, and other graphs. The element graphs themselves are sets of nodes and edges. The semantics of an edge is given by its direction and its label. The label of a node is represented within an ellipse.

Two basic modelling techniques are used in CSDL to organize knowledge. The first of them orders knowledge along a number of "hierarchical dimensions"; it is, in fact, a kind of classification scheme. Elements of the CSDL so-called "ISA (is a) hierarchy" are both entities and "frames". The notion of a frame is basic to the second modelling technique; it is defined as a high-level data structure that describes the characteristics and behavior of an abstraction or a stereotyped situation.

Graphically, frames are depicted as rectangles, surrounding a predicate (concept) and its arguments. The arguments are linked to the predicate by edges representing and labelled as roles or cases. The list of case labels originally proposed was:

agent (a)	the instigator of an action
object (o)	the recipient of an action
source (s)	the location or the controller of an object before the action
destination (d)	the location or the controller of an object after the action
time (t)	the time of the action
result (r)	the result of an action or function
property of (pr)	the object that has a property
value (v)	the value of a property
argument i (argi)	the i-th argument of a function

Additional or different cases may be used for different applications and cases have already been extended for the purpose of modelling general system requirements [MIT 80].

Fig. 3:2 shows a simple-frame named TEACH with 'teach' as its action predicate related to arguments "filled" by the variables x, y, and z of type INSTRUCTOR, COURSE, and SEMESTER, respectively. The frame TEACH is the intensional description, with unassigned variables x, y, and z linked to generic concepts. TEACH does not assert anything about any actual teaching events, but a particular extension with values of real occurrences will be stored in a database as a data structure: e.g., in a relational database management system, TEACH (INSTRUCTOR, COURSE, SEMESTER) would define a set of tuples that represent occurrences of actual teaching events.

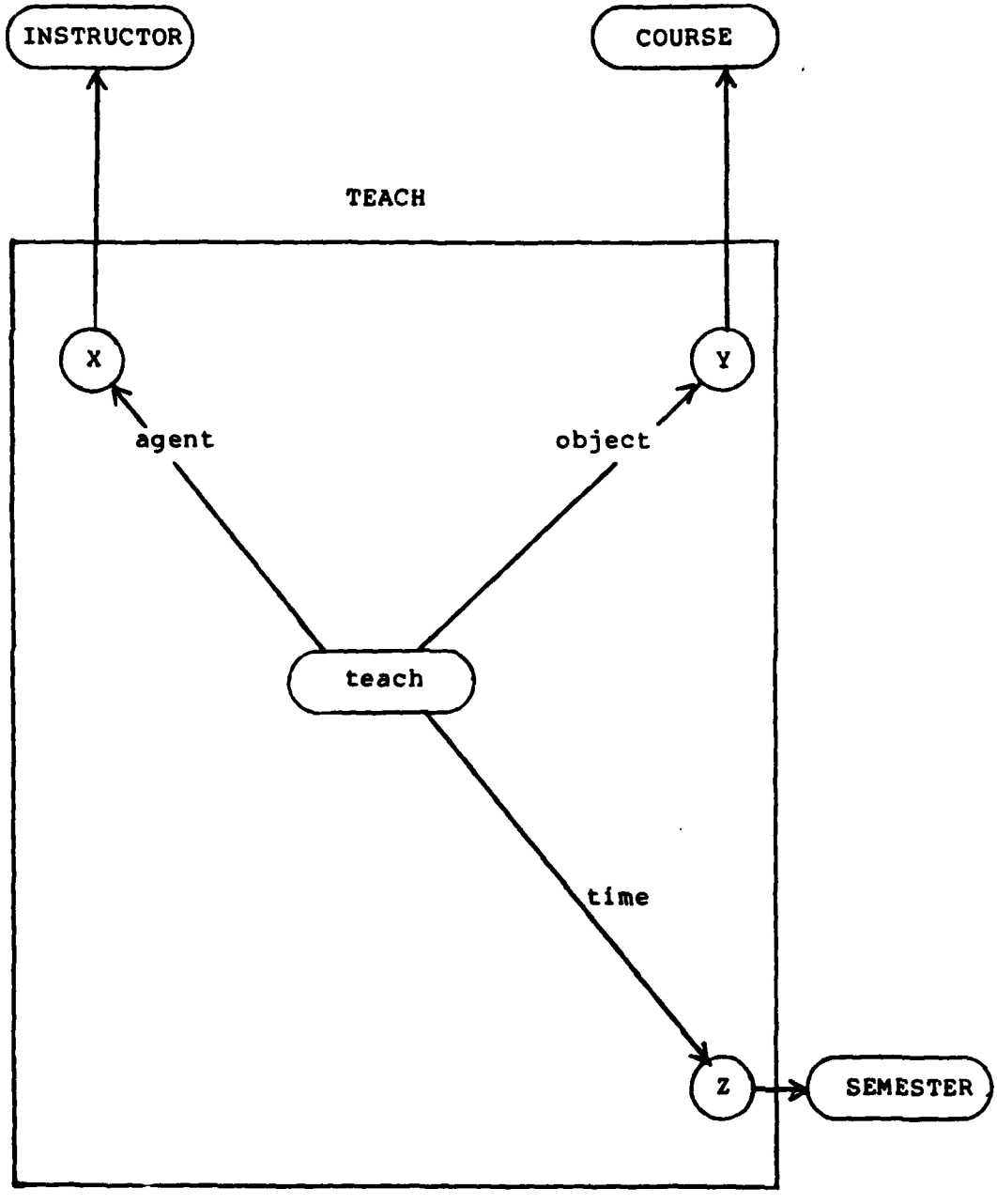


Figure 3:2 Instructors 'Teach' Courses During Semesters

The frame concept can be used to identify, represent, and specify events that are defined in a classification scheme. The Identification of events may start from scanning each action as it is encountered in the information environment, because actions are basic for relating different events. In order to identify events it is necessary for each action to be checked with the case list shown above - i.e., to identify its agent, object, source, destination, time, result, property, value, and arguments.

For example, Fig. 3:3 shows a frame named WORK with 'work' as its action concept, that relates operators to the machines on which they worked, and to the date of the work. Each generic concept may be complex and described by a set of properties and/or by a separate frame (e.g., NAME and SSN of the OPERATOR concept).

The classification hierarchy can be directly represented in CSDL by the "ISA hierarchy". For instance, the OPERATOR concept can be defined in terms of the "ISA hierarchy" shown in Fig. 3:4. The corresponding hierarchy of the WORK frames is also represented in Fig. 3:4. The frame WORK is now a specialization of a more general WORK frame, which allows representation of information on "who was working, when, and for how many hours". A most important feature of the CSDL "ISA hierarchy" is that properties of instances of "higher" generic concepts are automatically "inherited" by the instances of "lower" ones, as exemplified by the PERSON/EMPLOYEE/OPERATOR hierarchy.

Simple frames can be linked together to model more complicated situations. For example, Figure 3:5 shows the conjunction ENROLLMENT of two simple frames: TEACH and RECEIVE-GRADE.

In the original CSDL language, links between two frames do not explicitly indicate the order of their usage. The extended CSDL language [MIT 80] allows the user explicitly to represent dynamic properties of a system by expressing the precedence between two frames linked by so-called invocation arguments. In this way semantic methods in their extended form can be used for modelling control structures.

3.5 CONCLUSIONS ON DATABASE DESIGN

The problems of database design range from application domain analysis to system-dependent implementation. No general design process has been arrived at to integrate existing methodologies that concentrate on only a few aspects of the design process. In particular, it is common in the current state-of-the-art that early design stages are to a great extent left to the designers themselves, who therefore must rely heavily on their experience and intuition. An integrated approach to database design needs a requirements analysis methodology that extracts design information from users that is both accurate and complete. It should include such important points as:

- identification of phenomena in an application domain;
- techniques for data collection and classification;
- specification of the data to reside in a database.

Accordingly, the Project Team concentrated on a variety of methodologies that addressed the early stages of the information system and database design process. A review of these methodologies is contained in the next section of this Report.

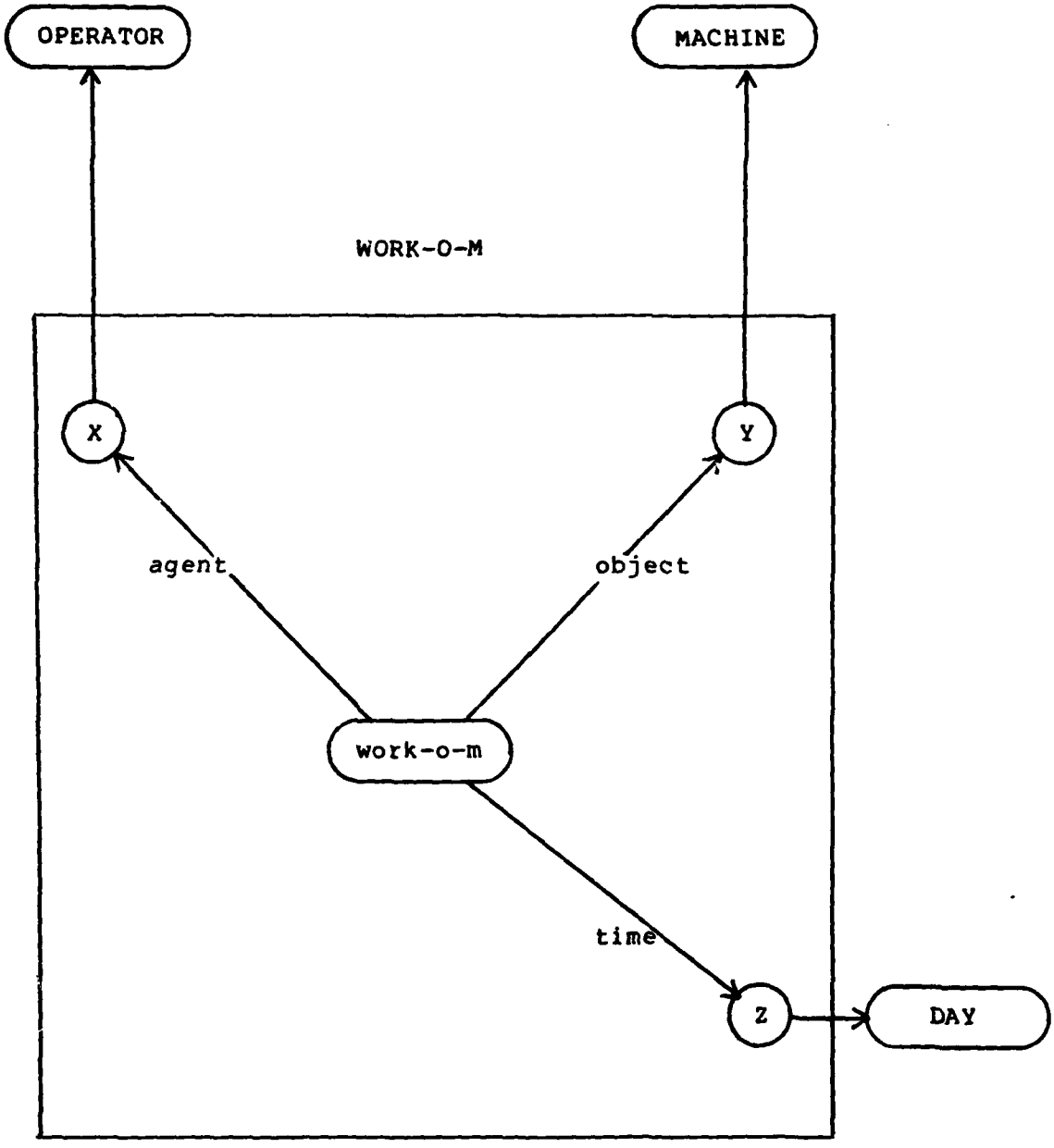


Figure 3:3 Operators 'Work' on Machines

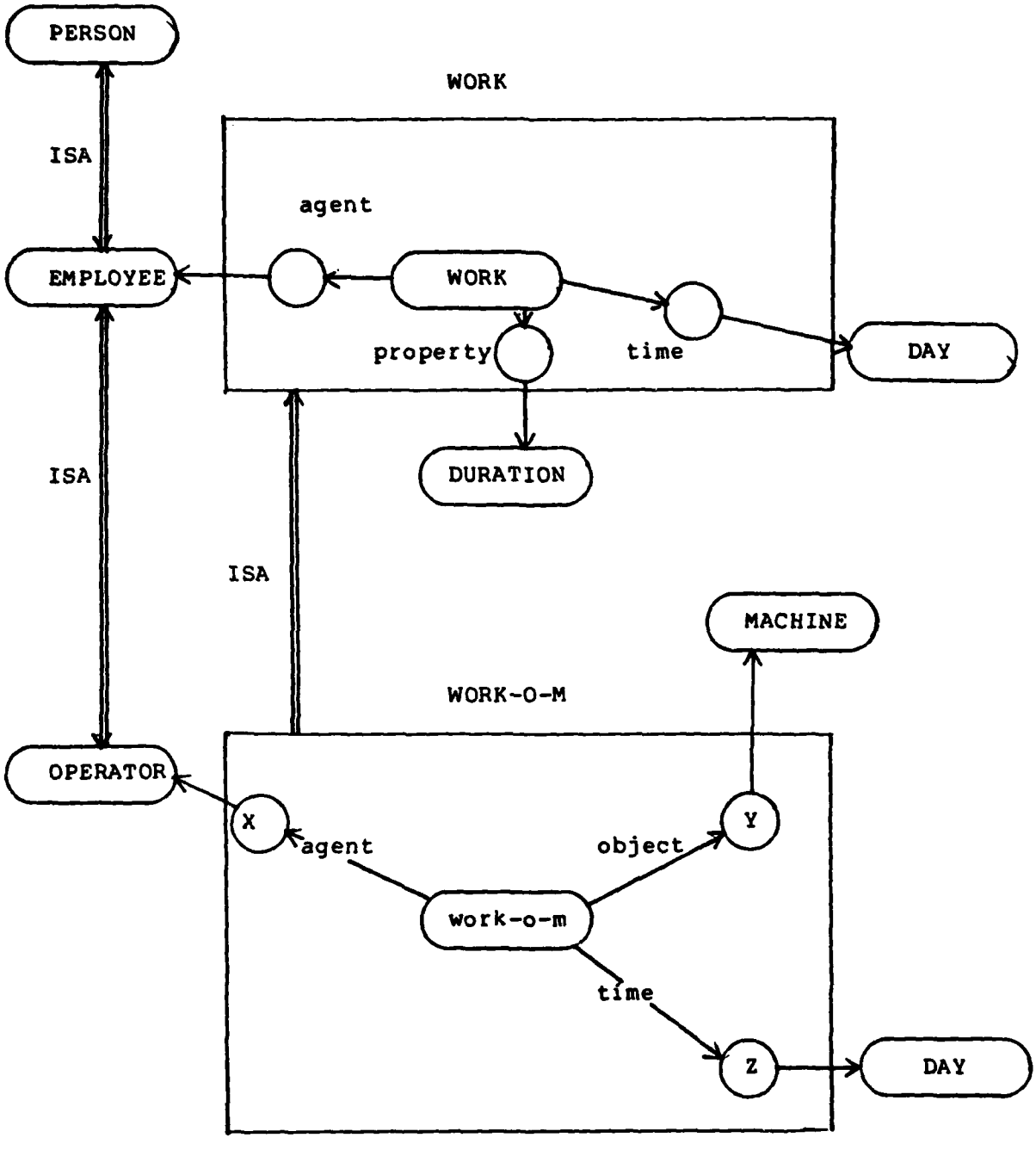


Figure 3:4 Hierarchy of Concepts and Frames

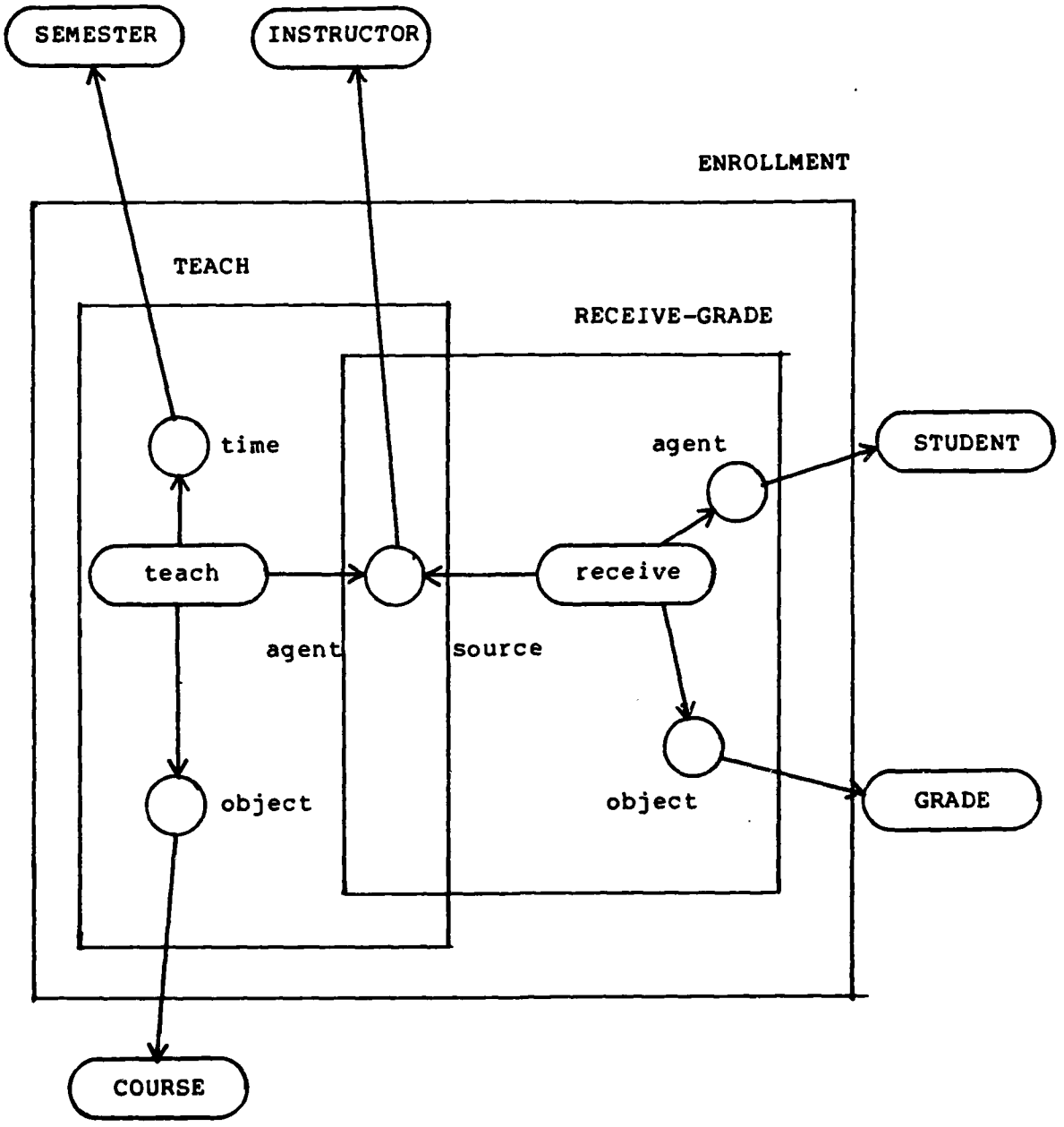


Figure 3:5 Conjunction of Frames

SECTION 4

A REVIEW OF REQUIREMENTS ANALYSIS METHODOLOGIES

This section briefly describes techniques currently in use or that have been used or that have been suggested as requirements analysis methodologies. For each technique, the description includes an overview of the approach, a methodological outline, and a list of advantages and disadvantages.

4.1 FINITE STATE MACHINES (FSM)

Finite state machine (FSM) models have been used as an intellectual framework for recording the development of software systems. A finite state machine is a sequential processing device which has no memory but has the control capability necessary to sequence from one processing state to another and to perform internal computation at each state. There is a stated upper bound on the number of states; the sequence of states starts at some predefined initial state.

A finite state machine may be graphically portrayed in state transition diagrams, where circles represent states and arrows indicate transition paths to the next state. Each transition is labeled, giving its input or output.

When the FSM model is used in recording requirements analysis, the required system is viewed as a finite state machine in which the system can be represented as any one of a finite number of states at any one time.

These states represent:

- o control processes (decision making);
- o conditions (information requiring storage/retrieval);
- o functions (tasks, processes or other executable processes).

The input and output from a state can be:

- o prompts - notification stimulating transition to the next state;
- o data - retrieved or stored items.

The FSM model is of greatest use to the analyst at the point of structural analysis and model construction. The analyst can examine the graphed system using the theoretical mathematical properties of FSM's to obtain a better understanding of the system environment and relationships between activities of the organization.

The use of finite state machines has several advantages over other approaches:

- o it does not obscure data flow;
- o it allows greater completeness and consistency checking;
- o it provides a structured way of considering semantic relationship and dictates a systematic approach to requirements decomposition and analysis;
- o it represents real-time processing well;
- o it provides a means of describing flow of data, control, and functions in an easily-understood diagram.

Its limitations stem primarily from a lack of an explicit methodology. The analyst has little assistance in avoiding ambiguity or in knowing what is a good representation.

Finite state machines are useful in physical design and procedure development stages of the target system. The model focuses on the procedural and control aspects of the target system. Its primary use is as a mental tool for the analyst, as it communicates graphically.

4.2 PETRI NETS

Petri nets, a technique developed in 1970 by Carl Adam Petri [PETR75], provide a means of depicting state machine representations of systems that have nondeterminate properties. Petri nets are composed of:

- o a finite set of places (conditions in requirement analysis);
- o transactions (events);
- o an alphabet of operators;
- o a starting place; and
- o a set of final places.

In requirements analysis, Petri nets model system operators which are decomposed into conditions and events. The model may be represented as a directed graph with labeled circles representing conditions and labeled vertical bars representing events. Directed arrows indicate the input or output interfaces of an event. An event is fixed if all the input conditions are satisfied. Tokens, represented by a solid black dot, show satisfied input conditions.

Initially the starting condition contains a token. When a condition occurs so that an event can occur, the token is moved from input to output, and if branching occurs at the output, two tokens will be generated. Thus, afterwards tokens are distributed through the Petri net according to the system execution sequence.

Selection rules defined for system operations determine event execution represented by the distribution of tokens to conditions, known as marking. The tokens are moved from input conditions to output conditions for the selected event. Repeated application of all selection rules produces a simulation of all possible event sequences.

Balkovich and Engelberg [BAL76] proposed the use of Petri nets, with formal logic and simulation for stating requirements and then analysis. The technique, involving a machine processable Petri net model, was tested in benchmark specification for a ballistic missile defense system. This research revealed the following advantages and disadvantages of using the modeling technique for requirements analysis.

Advantages:

- Petri nets can adequately describe operational rules of the system;
- Petri nets can be used as a simulation model that can verify that stated performance requirements meet system objectives.

Disadvantages:

- Petri nets are not very helpful in deciding the degree of decomposition necessary to describe the system adequately;
- Petri nets can become very large and unwieldy. A large system would need automated aids for generation and analysis;
- Petri nets cannot adequately describe performance requirements, though they are useful in simulation;
- Petri nets are inadequate as a System Specification Language.

Petri nets are primarily useful in the physical design and procedural development stages of target system development. They focus on the control and procedural aspects of the target system.

4.3 VERIFICATION GRAPHS

Belford, Bond, Henderson, and Sellers of Computer Science Corporation proposed the use of verification graphs methodology to decompose systems requirements in order to ensure an accurate and complete specification [BEL76]. Verification occurs in the design process, between the "system requirement engineering" and "data processing subsystem engineering" steps. The methodology produces an updated accurate and complete system specification.

The analysis occurs in three phases:

- decomposition of the functional and performance requirements into Decomposition Elements (DE's) -- DE's are statements of functional requirements and associated performance characteristics, developed from the system specification;
- static analysis of performance characteristics through use of an automated DE translator that generates verification graphs to determine DE allocation consistency;
- dynamic analysis of DE's in response to system changes as a result of verification graph analysis or simulation.

Within each phase, the analyst performs two steps:

- vertical and horizontal decomposition of the specification to define all processes and relationships;
- verification, via trace, of all decomposed elements.

The analyst must list the following information for each decomposition element on a System Specification Language Form:

- input;
- process;
- output;
- conjunctive connectivity;
- performance requirement;
- disjunctive connectivity.

This data is entered into the translator database. The translator, an automated tool, performs the following functions:

- maintains a database of the decomposition elements, etc.;
- syntax analysis;
- verification graph generation;
- verification graph analysis;
- report generation.

The verification graph, a directed graph, replaces the traditional flowchart with nodes representing functional processes and the stimuli/responses (i.e., input/output) forming the links (di-links). A digraph connectivity matrix may then be generated to map each node in the system to any other interfacing node. The matrix is generated by making a row and column entry for each system node with "1" in the ith row and jth column representing a connection from the ith predecessor node to the jth successor node. A zero entry means no connection or dependency.

The authors of the technique cite these advantages which their experiments demonstrated:

- early identification of requirement definition errors;
- exhaustive tests for completeness, consistency, ambiguity, and closure;
- means to detect errors easily;
- understandable to unskilled users;
- computer-assisted aids for automatic graphing, problem reporting, and simulation modeling.

The technique has the following disadvantages:

- decomposition accuracy is tightly coupled with the grammatical complexity of the original system specification;
- the decomposition methodology encourages over-specification of requirements;
- the methodology requires close contact between the decomposer and specification writer and system users.

The technique is a valuable check for detecting errors and ensuring that the developer understands user requirements but should not be used as the only means of structuring requirements.

4.4 N-SQUARED-CHARTS

The Systems Engineering and Integration Division of TRW Defense and Space System Group developed N-squared-charts as a supplement to other structured approaches to give equal weight to interface tabulating definition, design, and analysis [LAN77]. N-squared-charts are intended to describe system interfaces in an easily understandable graphic format.

The basic N-squared-chart is a square matrix of block diagrams, where allocated functions are on the diagonal, inputs to the function are on the vertical axis, while outputs are on the horizontal axis. Interface squares (nondiagonal blocks) represent unidirectional interfaces between two functions. All functions and interfaces are labeled.

There may also be symbolic additions and variations which clarify interface requirements:

- o substitution of numbered circles for interface blocks;
- o elliptical symbol to indicate a conditional interface;
- o circle with more than one arrow to show a multiple bus interface;
- o shading to identify major interfaces;
- o a larger circle to show two functions with control loop interface;
- o a torn block to indicate continuation;
- o dotted-line circles linking functions with a loop interface;
- o a circle with inner arrows to designate steps on time-linear sequencing of functional interfaces.

To develop an N-squared-chart, the analyst:

- o lists all functions;
- o lists all interfaces to each function;
- o lists the details of the interface, including conditions that must be satisfied at the interface prior to data transfer;
- o plots the chart.

The chart helps the analyst obtain the following information:

- o missing interfaces, e.g., stand-alone functions;
- o inconsistent interfaces;
- o overly complex interfaces that can be simplified;
- o critical element interfaces - one function with interfaces to many other functions;
- o tightly related groups - functions with exclusive interfaces;
- o loop conflicts or inconsistencies;
- o time sequenced activity inconsistencies.

The approach has several benefits:

- o it forces systematic and structured evaluation of all system interfaces including the relationship between people, organization, and functions;
- o the completed chart communicates interface details very clearly;
- o the approach considers non-software requirements such as personnel management, project planning and scheduling;
- o it is useful in spotting inconsistencies and incompleteness in a requirements statement.

It has two problems:

- o there are not clear rules for spotting incomplete or inconsistent statements;
- o N-squared-graphs become large and unwieldy for systems.

The N-squared approach is most useful when used as a supplement to some other structured requirements analysis approach where automated analysis aids are available.

4.5 FUNCTIONAL FLOW DIAGRAMS AND DESCRIPTIONS (F^2D^2)

The Functional Definition Group of RCA developed functional flow diagrams and descriptions as a method of defining, allocating, controlling, and auditing requirement design and operational development [LUR77]. It can be used as both a design tool in system definition, and as an audit tool to verify and validate a developed system.

The F^2D^2 methodology has five basic components:

- o top down approach to system definition;
- o tiered flowcharting technique for representing functional allocations;
- o time and sequencing chart;
- o automated support tools to be used for analysis, program generation, test, etc.;
- o textual description of the completed requirement breakdown.

Using this approach, a system engineer performs a "top-down" functional decomposition of the system, using five steps called tiers. Each step or tier increases the level of descriptive detail. Each tier is partitioned into functions that represent the system at that level of detail. The five basic tiers are:

Tier 0, Mission Definition Tier

describes the top level functional flow of the system; the underlying subsystem details are hidden;

Tier 1, Subsystem Tier

the analyst allocates each function to subsystems or elements; defines the methods necessary for the system to meet the mission requirements; and allocates functions necessary to make all operational actions;

Tier 2, Computer System Components Tier

the analyst uses the results of Tier 1 to allocate functions to system components such as computer programs, personnel or equipment;

Tier 3, Computer System Architecture

describes detailed operator tasks, hardware design, and computer program module design requirements;

Tier 4, Detailed Partition and Design level

the analyst details design requirements for individual program modules and routines and breaks the equipment groups into configuration items and detailed personnel procedures.

The analyst presents the results of all tiers in flow diagrams. After the flow diagrams are completed, the analyst develops Sequence and Timing Diagrams.

The authors of the F^2D^2 methodology based the decomposition and notation on a message communication system with all input or

output included in message data. They developed distinct symbols for each tier.

The Sequence and Timing diagrams which accompany the graphic depiction represent a needle and thread pass through the functional flows. Execution scenarios are plotted with functions tied together by interfacing data.

In addition to the flow diagrams, the analyst produces a textual description of each function, incuding:

- o abstract - statement of purpose;
- o input data - all messages initiating the function;
- o control data - data needed to control performance;
- o output data - results;
- o criteria - how the function operates;
- o allocation - references the requirements document or specification when it is generated.

The methodology has several advantages:

- o enables the requirements engineer to select from alternate choices of function sequence, determine the best system design approach, make tradeoffs among allocations;
- o gives enough flexibility to handle most projects;
- o appears to reduce software requirements errors;
- o does not require extensive training;
- o emphasizes the man-machine interface.

The primary disadvantage of the methodology is that the automated tools necessary to make the technique work are the property of RCA and are not sufficiently generalized to be useful to non-RCA projects.

4.6 HIERARCHY PLUS INPUT PROCESS OUTPUT (HIPO)

HIPO is a design aid and documentation tool developed by IBM's Systems Research Institute [IBM75]. It uses top-down design and hierarchically structured representations. A HIPO package consists of diagrams which describe the system product, management guidelines for developing and maintaining the HIPO diagrams, and optional automated support tools. The objectives of the approach are:

- to provide a structure by which the system's functions can be understood;
- to state the functions rather than specify program statements;
- to provide a visual description of each input and output.

The package uses three sets of diagrams:

- visual Table of Contents - shows the structure of the package and relationship of the hierarchical functions and serves as a graphic index;
- overview diagrams - give high-level description of major functions and list all input processes and output;
- detailed diagrams - lowest level requirements analysis diagrams and show the breakdown to specific input and output and one or more detailed subfunctions.

The management guidelines call for design reviews in three cycles: initial design, detailed design, and maintenance. Each cycle requires an updated set of HIPO diagrams. IBM markets a worksheet and a symbol template to increase the consistency of graphic symbols and graphic aids.

The system has the following advantages:

- it is a generalized methodology;
- it reduces errors by giving a formalized and standardized approach to requirements analysis;
- the system's graphics present a clear picture of the system as a whole, especially the documentation of I/O with each function;
- the system's use is easily understood even by non-DP people.

The system has two major disadvantages:

- there are no formalized methods of consistency or completeness checking;
- the automated support tools must be purchased or developed, or the users must rely on a manual process.

4.7 STRUCTURED ANALYSIS AND DESIGN TECHNIQUE (SADT)

SADT, developed by SOFTEC, Inc., provides both graphic documentation and a management approach to system development [SOF76, ROS77]. SADT is a total systems methodology emphasizing top-down decomposition, team effort, and systematic review at each developmental stage. In using the approach the analyst develops the diagrams interactively using the ideas of necessity, dominance, and relevance. The system has been used in a wide variety of applications.

The approach represents the system as a hierarchical structure of four-sided boxes, where each box represents one subset of the system. A set of boxes at the same level represents one view of the system. Each view is presented on a single page with arrows representing interfaces between the activities referred to in the boxes. All SADT models have dual sets of diagrams: one representing data, the other representing activities. The system provides a grammar of 40 syntactical variations to express decomposition. This grammar allows the user to:

- define the graph structure;
- build the box structure;
- show distribution subdivision and exclusion;
- increase readability.

The complete collection of diagrams provides three views: an analyst view, a management view, and an operational view. The top-level diagrams present an overview and the lower level diagrams give detailed information.

The system has several advantages, particularly when used in the context of strong configuration management:

- it supports communication of the system structure;
- it formalizes the decomposition process;
- it reduces requirements definition errors.

The system has these disadvantages:

- it does not represent activity sequences or data flows, so the analyst cannot show loops or performance requirements;
- it lacks formal rules for determining relevant necessary or obvious interfaces and the analyst may have difficulty establishing each entry's relevance and therefore may omit the obvious interface;

- it currently lacks automated tools to store and retrieve correlated data subsets;
- it lacks quantitative theorems to determine if the model is complete and consistent;
- all analysts and model users must learn the 40 syntax mechanisms to use the model effectively.

4.8 STRUCTURED SYSTEM ANALYSIS

A number of requirements analysis approaches are known under the generic name of "Structured System Analysis" or SSA [MYE78]. These techniques share certain qualities:

- most are not rigidly standardized;
- they use "bubble charts" to show data flow and transformation;
- they use structure tree charts to show system components with elaborate annotation to identify components and their relationships;
- they use top-down decomposition and Parnas' principles of abstract state machines and information hiding to develop the structure;
- many approaches ignore management guidelines;
- they use standardized data dictionaries to record and define data elements;
- the approach correlates with data structuring techniques based on Codd's relational data model [COD79];

Taken as a whole, these techniques have these advantages:

- the technique can be used throughout the project's life cycle;
- the use of a data dictionary makes it easier to maintain and update the diagrams and resultant design.

There are some disadvantages:

- lack of rigid requirements;
- failure to address all aspects of requirements analysis;
- limitations of the block structure approach which include:

- inability to model non-hierarchical structures like loops;

- difficulty of representing performance characteristics.

4.9 ISDOS/PSL/PSA

The Department of Industrial and Operations Engineering at the University of Michigan developed ISDOS (Information System Design and Optimization Technique) [TEI77]. This system aims to provide computer tools which would ensure consistency, traceability, and requirements allocation in Management Information Systems. The Michigan team reviewed existing systems analysis and requirements statement techniques and adapted the best features from several techniques. In addition ISDOS draws upon the relational model of data, particularly as used by CODASYL Development Committee.

The analyst records the target system requirements using the Problem Statement Language (PSL). The language describes a system in terms of objects which have properties with specific values and relationships between objects. The language recognizes 22 object types and 55 relationship types.

The PSL can describe eight major system aspects:

- o system input/output flow interaction between the target system and its environment;

- o system structure -- represents hierarchical relationships between system components;

- o data structure -- represents relationship among data as viewed by users;

- o data derivation specifies the way in which the target system manipulates or derives data;

- o system size and volume describes system size and factors influencing the processing volume;

- o system dynamics presents how the system behaves over time;

- o project management -- documents the current project to describe the target system.

The analyst enters the information about the target system expressed in PSL into a database by using a software package, the Problem Statement Analyzer (PSA). The PSA can produce a variety of reports; some it generates automatically, others on demand.

These reports fall into four types:

- o database modification reports - record changes to the database for error correction and recovery;
- o reference reports - give formatted data used as reference, including directories and dictionaries;
- o summary reports - present data collected in various ways, like STRUCTURE report which present hierarchies as a list of nodes or graphically;
- o analysis report - presents data helpful in project management.

PSL/PSA has several advantages or capabilities:

- o describes information systems regardless of application area;
- o records system descriptions in a computerized database;
- o incrementally adds, modifies, or deletes from the system description contained in the database and so controls revisions;
- o produces hard copy versions of all inputs and produces data useful for analyzing data relationships;
- o is available on a number of computer systems and transportable to others since it was written in ANSI FORTRAN;
- o is compatible with current procedures in many organizations including the U.S. military.

PSL/PSA disadvantages are:

- o language deficiencies do not allow the specification of procedural information necessary for real time constraint or the specification of processing steps and sequences;
- o natural language narrative information is still necessary for human readability;
- o the languge syntax is unfamiliar to non-computer personnel and requires extensive training;
- o analysis tools for optimization, simulation, and functional tracing are still being researched and developed.

The lack of analytical tools and of quantitative data on the economic benefits provided by PSL/PSA brings into question whether it has any advantage over the graphic methodologies used with automated documentation systems.

4.10 SYSTEM OPTIMIZATION AND DESIGN ALGORITHM (SODA)

J. F. Nunamaker developed a System Optimization and Design Algorithm (SODA), as part of the ISDOS project to generate a set of design alternatives [NUN76]. The Navy used SODA in the development of an integrated financial management system for the Navy Material Command Support Activity (NMCSA).

The SODA has two classes of input statement: augmented Accurately Defined System (ADS) problem statements and SODA System Statement Language (SSL) statements that describe performance data.

ADS is an NCR product that views the problem statement as:

- o describing the reports produced as output;
- o defining the system which produces this output from the input.

ADS has four elements:

- o forms describing the target system's inputs;
- o historical data needed for information system storage;
- o output identification;
- o descriptions of actions required to produce outputs and the conditions necessary for each action.

The computer-aided ADS used with SODA also includes automatic syntax validation, additional formatting rules and naming conventions, a data dictionary providing data elements for each process, precedence relationships among data elements, and linkings indicating each data item's reference.

The SODA SSL is a set of forms with which to gather volume and frequency input and output data.

The ADS statements supplemented by SODA SSL are input into the SODA System Statement Analyzers (SSA). The SSA produce:

- o syntax analysis of machine-readable SSL statements;
- o error messages;
- o a matrix recording the relationship of processes and data to output;

- input necessary for the SODA Generator of Alternatives (SGA).

The SGA uses:

- the SSA output;
- hardware file data, that is the machine characteristic;
- software file data.

The SGA produces recommended program module *groupings* and computes the expected processing time for each alternative design. The SGA breaks development of the alternative program groupings into two phases:

- it traces the sources of data items to identify the elementary processes involved;
- it groups the elementary processes.

In the last stage of work the SODA Performance Evaluator (SPE) simulates the system with this model to determine resource utilization and system performance data. The SPE produces:

- a list of hardware resources;
- identification of program groups;
- a file list with storage device assignment;
- program run sequences necessary to accomplish the requirements.

The system has these advantages:

- it provides information needed for system analysis including performance data;
- system facilitates early feedback on errors;
- system is helpful in validity, completeness, and logical consistency checking;
- system works with large systems development.

Its disadvantages include:

- resistance by personnel to forms-oriented procedure;
- lack of checking for semantic correctness.

4.11 HIGHER ORDER SOFTWARE (HOS)

Higher Order Software (HOS), developed by MIT's Draper Lab, uses an axiomatic approach to system decomposition [HAM76]. The methodology has six major components:

- application of a formal set of laws applied to the given problem;
- a specification language AXES intended to ensure that the problem statement is consistent with the formal laws;
- automatic analysis aids for checking static and dynamic constency completeness and correctness;
- system architecture described in virtual layers produced from output of above step;
- description of hardware needed;
- description of support systems.

The approach uses a limited finite state machine representation in which the target system is a hierarchy of elements expressed mathematically, and in which control is the defining relationship of the hierarchy. Each level is a virtual layer defining all input and output variables for that level of the system structure. The top layers give requirement definitions; the bottom layers describe the hardware interfaces. Each layer consists of a process tree of functions where each node specifies the relationship of one input element to a single output element which corresponds to that input.

According to the developers, five aspects of control can describe all the relationships between processes necessary to guarantee interface completeness:

- invocation of processes;
- access rights;
- precedence ordering;
- responsibility - which process produces the output;
- rejection - recognizing improper input.

The developers maintain that if the following six axiomatic conditions are met, then the system interfaces are complete:

- processor controls the invocator of the valid vunctions on its immediate lower level and only its immediate lower level;

- there is no member of the input space for which no output space is assigned;
- a processor controls the access rights to outputs of its immediate lower level and only its immediate lower level;
- a processor controls the access rights to inputs of its immediate lower level and only its immediate lower level;
- a processor controls rejection rights to its own input;
- a processor controls ordering for its immediate lower level tree and only its immediate lower level tree.

4.12 SOFTWARE FACTORY

The Software Factory is a package of software tools to support software development [BRA75]. The tools assist the engineer by providing the following capabilities:

- flexible yet disciplined and repeatable methodology for development;
- management visibility for tracking project development;
- ability to update requirements as the design evolves;
- verification and validation tools;
- reusable software development tools.

There are six major tools in the package:

- Factory Access and Control Executive (FACE) performs control and status gathering for all processors, supports the factory command language, integrates the processor with the system development database, and provides production library services;
- the Project Development Data Base consists of a software database and a project control database;
- Automatic Documentation Tool (AUTODOC) produces program and system documentation;
- the Program Analysis and Test Host (PATH) provides recording of program runs for static profile analysis reports;
- the Test Case Generation (TCG) provides for test data design;

- the Top-Down System developer (TOPS) allows succeeding logic levels to be modelled with calls to program stubs of the next level when it has not yet been designed;

- the Integrated Management, Project Analysis and Control Technique (IMPACT) provides documented managerial information on the project itself; it can produce schedule, resource utilization, and status reports at each development stage.

The Software Factory shares the advantages and disadvantages of other project development library systems:

- it integrates requirements analysis, configuration management, and software design;

- it provides good software capabilities for project management.

The greatest disadvantage is the limited support it provides for automated completeness and error checking.

4.13 STUDY ORGANIZATION PLAN (SOP)

The Study Organization Plan (SOP), produced by IBM [IBM61], integrates the approaches of several requirements analysis techniques. SOP is used to gather data for the information needs analysis of the entire organization.

The analyst uses an integrated set of forms to record information about the enterprise's information needs. For each activity the analyst completes two forms:

- the Resource Usage Sheet shows the cost impact for the activity and fits each activity into the larger environmental structure;

- the Activity Sheet breaks the activity into its major operations and presents them as a flow diagram with individual blocks representing various operations.

For each operation block on the activity sheet the analyst completes an Operation Sheet that shows relationships between inputs, processes, resources, and outputs.

Two other forms support the information contained in the Operation Sheet:

- the Message Sheet describes the inputs and outputs of the activity;

- the File Sheet describes a collection of messages, an information file and identifies which stored information the operation utilized.

As a last step the analyst organizes the information on the forms into a report in three sections:

- a General Section giving a history of the enterprise, industry background, goals and objectives, major policies and practices, and government regulations;
- a Structural Section containing a schematic model of the business in terms of products and markets, materials and suppliers, finances, personnel, facilities, inventories, and information;
- an Operational Section including flow diagrams and a chart showing total resource by operating activity. These charts show how the business responds to inputs, performs operations, and produces outputs.

Although SOP was among the earliest requirements analysis methodology to be developed and used, and although it is no longer used in its original form, it has been included here for several reasons:

- it is one of the only methodologies that attempts seriously to offer control of the earliest stages of information systems design, as can be seen from Fig. 5:1;
- it is one of the only methodologies that addresses the problem of analyzing the activities of the enterprise;
- its concepts are pervasive and may be found in other, later methodologies.

Only Wilson [WIL79] seems to probe as deeply into this area. His methodology, covering system design as well as requirements analysis, was discussed in Section 3.4.1.

SECTION 5

SUMMARY OF THE REVIEW OF REQUIREMENTS ANALYSIS METHODOLOGIES

5.1 COMPARISON OF METHODOLOGIES

Each requirements analysis methodology must deal with three aspects of the target system: data, procedure, and control. One point of comparison between the methodologies is the aspects upon which they focus, and how they describe those aspects.

The data aspect, and particularly that part that deals with data structures, concerns the most static component of the target system. Data are the material of the information environment that the system manipulates in order to perform an assigned task.

The procedure aspect deals with the way in which the system performs that task.

The control aspect determines when and why the system operates, i.e., what situations cause certain procedures to be executed.

Each technique described in Section 4 deals in some way and to some extent with all three aspects, but each technique has one, or perhaps two, aspects as its major focus, with the other aspects discussed as minor aspects within the primary aspect's description. This section shows the requirements analysis methodologies by the system aspect which is their main focus (some methodologies fall into two categories):

5.1.1 Data Aspect

SADT uses one of two sets of block diagrams to show data in a procedural context;

Structured Systems Analysis (SSA) approaches use a data dictionary to record and define data elements;

PSL/PSA specifies data structures as seen by the users of the system and data groups in information collections like documents.

5.1.2 Control Aspect

Finite state machines (FSM) show what input controls the transition from one state to the next;

Petri nets use the movement of tokens to show which conditions cause a sequence of conditions to be fixed;

2 2
Functional Flow Diagrams (F D) show various execution scenarios through the Sequence and Timing Diagrams;

Higher Order Software (HOS) uses a limited finite state machine representation with five axioms of control to guarantee interface completeness.

5.1.3 Procedural Aspect

Finite state machines use graphs to represent functions performed by the system;

Verification Graphs represent the functions or processes of the system as decomposition elements;

N-squared charts focus on the interface between functions;

Functional Flow Diagrams use a tiered flowcharting technique to represent the functional allocations;

HIPO provides three levels of diagrams for presenting the procedures or functions of the system;

SADT uses one of two sets of block diagrams to model system processes;

SSA approaches use several techniques to show the procedural flow of the system;

SODA focuses on procedures the system uses to produce reports, and produces recommended program module groupings;

PSL/PSA describes input/output flow, and procedures for derivation of data;

Higher Order Software models the processes of a system in a top-down structured fashion;

The Software Factory includes tools to control processors, produce program documentation, record program runs and develop procedures in top-down fashion;

SOP describes the organization's activities in terms of operations.

5.2 CONCLUSIONS

A review of requirements analysis methodologies, supported by critical reviews by Couger [COU73] and Taggart and Tharp [TAG77], suggests that concentration in the past has been on identifying subsystems within an organization: production, maintenance, management, etc., and plotting their data and data flows. Consequently, information system and database design has not been approached logically so much as empirically. This approach is no longer adequate because of the sheer size and complexity of many modern data resources or environments.

The comparison of the methodologies by the aspect that is their main focus reveals that comparatively few focus on the data or control aspects, and most focus on the procedural aspect. The data aspect seems to be particularly neglected because those methodologies that describe data, like SADT, focus on data flow, or how the data is manipulated and changed.

Only PSL/PSA discusses data structure and the relationship among data items. As described in Section 4.9, its approach uses a kind of classification to organize data input from the information environment, and its analytical method produces a picture or view of the real world of the information environment that can be used to support subsequent stages of the information system design. Even PLS/PSA, however, depends to a certain extent on the availability of data input in a useful form.

The Study Organization Plan (SOP) is possibly the only methodology that deliberately and specifically attempts to capture and make explicit the data input from the real world. It is significant that SOP dates from a time before systems analysis had to concern itself with highly complex data handling procedures.

We can see from examining the chart showing the stages of system development (Fig. 5:1) that current requirements analysis methodologies deal more with expressing the requirements analysis than with determining the requirements themselves. That is, with the exception of PSL/PSA and SOP discussed above, the methodologies support the logical design phase and subsequent phases, with the expectation that input from the real world of the information environment is either already available or easily accessible in a form suitable for use by the methodology.

We therefore see the need for a methodology that will:

- collect and analyze the data as used in the target system or organization;
- organize the data in a structured and meaningful way to describe the existing organization;

- organize the data in a structured and meaningful way in terms of the new information system (i.e., logical and physical structure).

The requirements analysis methodology should provide a tool that will ensure the ultimate design is based on objective analysis of the information environment, rather than, as is now too often the case, on the analyst's ability to perceive the user's data needs by intuition.

In the light of the review of database design methodologies and specifically requirements analysis methodologies contained in this and the previous section, we felt justified in proceeding with an examination of the feasibility of facet analysis and faceted classification principles and structures.

	Documenting Existing System	Logical System Design	Physical System Design	Construction	Test & Conversion	Operation	Maintenance & Modification
Finite State Machines			<--------->				
Petri Nets			<--------->				
Verification Graphs			<------>				
N-squared Charts			<------>			<---->	
$F^2 D^2$			<--------->				
HIPO		<--------->					
SADT		<--------->					
SSA		<--------->					
HOS		<--------->					
Software Factory		<--------->					
SODA w/ADS		<--------->					
PSL/PSA	<--------->						
SOP	<--------->						

Figure 5:1 Comparison of Requirements Analysis Methodologies

SECTION 6

CLASSIFICATION THEORY AND FACET ANALYSIS

6.1 INTRODUCTION

A model and a methodology for the description of highly complex data elements already exist in a discipline not consulted frequently as a source of principles and techniques by information system designers: the discipline of information science, and its part predecessor and parallel discipline, library science. One of the problems long faced by librarians and information scientists is that of describing documents on often highly complex topics in a way that is explicit of all the aspects and parts, and yet is simple enough to be applied easily and to allow efficient ordering of lists and retrieval from databases. For example, a document on THE EFFECT OF THE BLACK DEATH (PLAGUE) ON LABOR MOBILITY IN THE ENGLISH WOOL TRADE IN THE MIDDLE AGES may be of interest to:

- o epidemiologists
- o economic historians
- o the wool industry
- o general medieval historians

and may be sought by any of them by a variety of key concepts, e.g., BLACK DEATH, LABOR MOBILITY, WOOL TRADE, and/or ENGLAND, etc.

The principal solution developed by the library and information science profession has been classification, expressed in a variety of ways. There are several kinds of index language, but the most efficient are those that acknowledge the principles of classification as they have been enunciated and elaborated over the last hundred and, in particular, over the last forty years. The most sophisticated of these approaches, and the one that seems to offer the most appropriate solution to the problems discussed in Section 2 and to come closest to the kind of organizing principle called for at the end of Section 5, is faceted classification, whose origin, development, and structure are described in this section.

In order to appreciate the richness and flexibility of modern (faceted) classification structures it will be necessary to review their bibliographic origins. It should be pointed out that the bibliographic environment that supported the development of synthetic and faceted classification theory is a highly complex one, which is much closer to the complexity of the information environment of the enterprise than is the world of the scientific taxonomies often considered the model for general classification. The distinction will become clear below.

6.2 EARLY CLASSIFICATION SCHEMES

The methods developed in the nineteenth century to organize knowledge in libraries, on shelves and in catalogs, reflected the slowly developing hierarchical pattern of knowledge itself as it had existed up to that time. Early library classification schemes reflected the taxonomic structure of the scientific and philosophical classifications of the day. Even when libraries based their classification on nonphilosophical, practical schemes for books like that of the Paris booksellers, the model was still very broad, simple, and discipline based [EDW59].

Methods for organizing subject catalogs and indexes were similarly naive. In the mid nineteenth century Panizzi in the British Museum proposed an "Index of Matters" based on catch words taken from the titles of books. Crestadoro [CRE56] proposed a more detailed index that anticipated Luhn's computerized Keyword-in-Context (KWIC) index [LUH59] by a hundred years, and in 1864 it was in use in the Manchester Free Library catalog. Cutter's "Rules for a Dictionary Catalog" [CUT76] provided the first comprehensive system for alphabetical subject cataloging. It was the basis of the subject headings list used in the Library of Congress. But Cutter was influenced by the contemporary climate of knowledge, and his subject headings are simple to the point of being simplistic, and the references that connect them (for example, BIRDS see BIRD MIGRATION) provide an equally simplistic pseudo-classification.

A typical library classification of the late nineteenth century was an enumeration in a simple hierarchy of all the conceivable topics about which books in the collection might be written. The rules governing the division into hierarchies were derived ultimately from Aristotle and his commentators [SAY18]:

- that division proceeds from general to specific;
- that it should be gradual;
- that it should be by useful characteristics of division;
- that the resulting classes and subclasses should be in mutually exclusive and collectively exhaustive arrays.

For example, Dewey's Decimal Classification (DC) [DEW76] divides all of knowledge into nine broad areas, preceded by a Generalia class that contains general works like encyclopedias and general bibliographies. Each broad area, like 300 SOCIAL SCIENCES, is further divided into the constitutent disciplines: 310 STATISTICS (a fundamental tool), 320 POLITICAL SCIENCE, 330 ECONOMICS, and so forth. Within each discipline, DC divides either into subdisciplines (or into aspects of the discipline), each of which reflects different characteristics. In 370

EDUCATION, the form subdivision 371 GENERALITIES OF EDUCATION represents the characteristic of problem or activity, whereas 372 ELEMENTARY EDUCATION is the first of several subdivisions that represent the characteristic of person, the educand. More and more detailed subdivision is represented by increasingly lengthy notation.

Sometimes in DC, frequently repeated sets of subdivision may be given distinctive notation, with instructions for its use in the main scheme of arrangement. For example, CIVIL ENGINEERING is given the notation 624, EDUCATION 370, and ENCYCLOPEDIA -03. Thus, 624.03 represents AN ENCYCLOPEDIA OF CIVIL ENGINEERING and 370.3 represents AN ENCYCLOPEDIA OF EDUCATION.

It was apparent even in the years immediately following the publication of systems such as those of Dewey and Cutter that these systems solved only the problems of the past. The present, not to mention the future, already held problems that they could not solve.

6.3 THE BEGINNING OF ANALYTICO-SYNTHETIC CLASSIFICATION

In 1895, Paul Otlet and Henri La Fontaine called a conference in Brussels at which the Institut International de Bibliographie (IIB) was founded; it is today known as the Federation International de Documentation (FID). One of the principal responsibilities of the IIB was to develop methods for the organization of nonbook material that was appearing in increased volume and in much more complex form. Otlet and La Fontaine realized that no purely enumerative system could ever attempt a complete listing in full detail, nor would it be satisfactory to use an enumerative system of "container headings", since each heading could not describe the complex material specifically and would probably contain too much material for easy browsing.

Otlet proposed a development of Dewey's Decimal Classification that would allow the extension of many aspects, the addition of still more, and, most importantly, the use of punctuation marks as notational devices to introduce as many aspects of description as would be needed to identify any specific item and to assemble them into a single decription [HOP07]. The resulting classification scheme later became known as the Universal Decimal Classification (UDC) [UDC33] and is used widely in Great Britain and Europe, mostly in special libraries and information services. It has also been used in special projects in the United States.

UDC was the first of the classification schemes known as analytico-synthetic, by which complex subjects could be analyzed into their constitutent aspects and a full description synthesized from those constitutent aspects so that the full description, or any subset of it, could be searched for to locate hitherto unrelated documents.

A full description of UDC would be out of place here, but an example may be helpful:

1979 YEARBOOK OF CAREER OPPORTUNITIES FOR WOMEN IN SCIENCE AND TECHNOLOGY IN GREAT BRITAIN

would use the components:

371.048	VOCATIONAL GUIDANCE/CAREERS (originally from DC 370)
-055.2	WOMEN
:	relational sign
5	SCIENCE (from DC 500 SCIENCE)
/	extension sign
6	TECHNOLOGY (from DC 600 TECHNOLOGY)
(42)	GREAT BRITAIN (originally from DC)
(058.1)	YEARBOOK (an extension of DC -05)
"1979"	1979

to make:

371.048-055.2:5/6(42)(058.1)"1979"

The numbers are like nouns or verbs; the punctuation marks are like conjunctions or prepositions indicating relationships or functions. Thus, all documents in the social sciences about women will contain -055.2; all documents about Great Britain will contain (42) - except its own history, which has a special number.

These features of UDC have been used in computer-based information storage and retrieval systems in the U.S., for example Freeman and Atherton [FRE68], and Caless and Kirk [CAL67].

In the first decade of the twentieth century, parallel to the development of UDC, J. O. Kaiser [KAI11] proposed a structurally similar solution to the problems of alphabetical subject headings: the use of a formula to assemble natural language terms in an order that would not only be consistent but would also be useful - the most important component would stand first, with other terms arranged in a decreasingly significant order behind it. That is, he proposed that what he called CONCRETES would stand first, followed by PROCESSES, and then by PLACES. He offered further refinements, but this is the basis of his scheme. His ideas were neglected for a generation, but later became the foundation (along with other ideas drawn from classification

theory) for modern indexing methods as used in the British National Bibliography and the British Technology Index.

Seventy years ago, classification theorists were already aware of the need to represent more complete composites of information in their classification and indexing schemes. However, a closer look at their debates reveals almost opposing views of what such composites were and what they meant to classification and indexing.

Otlet's extension of Dewey's classification scheme acknowledged that it would be impossible to predict and enumerate all complex topics found in documents, and that therefore a system had to be devised to allow necessary descriptive syntheses to be made when necessary. Indeed, Bliss, already at work on his Bibliographic Classification in 1902 [BLI10], was to assert later (in his principle of relativity) that a classification scheme should allow descriptive organization and synthesis not only in terms of the literature of a subject, but also in terms of user orientation. A British contemporary, J. D. Brown, also acknowledged a user orientation in his Subject Classification which ran counter to the customary discipline-based pattern of classification schemes [BRO06].

In a larger context, these men were responding to the problems often perceived by the owner of a personal library: should the books be arranged by general (accepted) discipline, or by problem, as customarily encountered by the user (or even by frequency of consultation). And what guarantee is there that the order chosen will remain constant even for the next project or consultation?

However, some contemporaries of Otlet, Kaiser, and Brown were classification theorists who maintained that classification schemes should include existing composites as full and explicit precoordinations, as they appeared in books. E. Wyndham Hulme [HUL12] wanted what he called "aggregates" like HEAT, LIGHT, AND SOUND (i.e., heat-and-light-and-sound) to be made the basis of classifications of books. Hulme's ideas reflected the basis of "literary warrant" for the inclusion of topics that lay at the heart of the Library of Congres classification scheme, then in its first stage of development. While it is true that Hulme, and others like him, recognized the existence of user-oriented syntheses in the literature itself, his proposals actually inhibited the development of systems that could lead to new demands for different syntheses for different users or user environments. Indeed, Hulme actually criticized such systems as "theoretical" and out of touch with the reality of the world of books.

Later developments, particularly since 1950, have followed Otlet and Kaiser rather than Hulme. Index languages used in information storage and retrieval systems and in systems for the selective dissemination of information have all chosen the synthetic model in one way or another.

6.4 CLASSIFYING FROM MULTIPLE PERSPECTIVES

T.S. Kuhn's idea of the emergence of a new paradigm to account for phenomena that are anomalies in terms of the existing paradigm [KUH70] is particularly apt to the developments in classification that gave rise to synthetic classification, even to his conclusions that the new paradigm will permit predictions that are different from those derived from its predecessor, and that the two paradigms will exhibit some fundamental incompatibility.

The situation facing indexers and classificationists at the turn of the century, and the development of synthetic classification and indexing may be explained partly in terms of Kuhn's theory of scientific revolutions. Subject classification of books on shelves for location or for browsing was insufficient for full exploitation of a collection; subject indexes or catalogs were needed as well. Subject classification schemes that are simple, enumerative hierarchies might continue to seem adequate for shelf classification, where only one "best" location is chosen for each title, and where the absence or confusion of subordinate detail may go unnoticed. But the application of such a scheme to an edited collection of papers where title surrogates may be included for any needed aspect quickly reveals the inadequacy of the scheme.

In Kuhn's terms, the paradigm of discipline-based hierarchical classification that had been accepted and tested for well over a century now faced problems and anomalies beyond its power to resolve. In response, a number of new paradigms - or at least a number of approaches to a new paradigm - were proposed. Just as Kuhn notes about the incommensurability of competing paradigms in science, much of the new paradigm incorporated vocabulary and apparatus from the traditional paradigm, but in the new paradigm the old concepts were used in a different way; also that the proponents of the old and the new paradigms looked at the same world in different ways.

In the world of classification, this has meant a view of classification and indexing that is rooted in hierarchical, deductive organization of knowledge (the enumerative approach) and also a view of classification as a set of parts to be assembled at need (the synthetic approach). Ironically, the enumerative approach is still instinctive for many Americans because of a single innovation that occurred just as the new approaches were expounded: the development of the Library of Congress Classification. This scheme and the accompanying catalog card service have perpetuated the traditional paradigm and inhibited favorable discussion of the competing paradigm until comparatively recently.

6.5 FACETED CLASSIFICATION

Faceted classification is a special kind of synthetic classification. Synthetic classification allows combination of component parts of any classification schedule, even an enumerative one. Faceted classification draws component parts from special lists (facets) which derive from the application of single, specific characteristics.

In order to understand the basis of much index language development on the synthetic model, it is necessary to examine the seminal work of S. R. Ranganathan [RAN58] in the field of faceted classification and the extension, revision, and rationalization of the analytico-synthetic systems of Otlet, Kaiser, and others.

Ranganathan pointed out that enumerative systems, and even analytico-synthetic systems like UDC that were based on enumerative systems, would always be inadequate because their very structure prevented them from specifying all possible details and from offering a sufficiently flexible combination of details. Further, he pointed out that the rigidity inherent in the enumerative structure (basically a two-dimensional structure as are all taxonomic classifications) meant that often they must ignore or modify the logical rules of classification.

Ranganathan's proposed Colon Classification, published first in 1933 [RAN33], explored a new set of dimensions of classification theory and practice. Each class or subject area was divided into aspects or "facets", each the result of the application of a single characteristic. So the class of BUILDING AND CONSTRUCTION might be divided by the characteristic of BUILDINGS to individualize kinds of building, by the characteristic of BUILDING MATERIAL to produce a list of all materials (entirely independently of kinds of building), and also by the characteristic of BUILDING OPERATION to produce a list of building operations. Any facet may be further divided into levels, for example BUILDINGS into KINDS OF BUILDINGS, PARTS OF BUILDINGS (HINGES). Alternatively, a facet may be divided into mutually exclusive subfacets that offer parallel arrangement by different characteristics, such as PEOPLE GROUPED BY AGE and BY SEX. Synthesized descriptions of complex topics could then be assembled from components drawn from each of these facets: SCHOOLS: HALLWAYS: DOORS: HINGES: BRASS: REPAIRING. These are simple examples to support this explanation; frequently a faceted classification has a much more complex structure.

Ranganathan's development of his own scheme [RAN33], based on an analysis of all the facets he had identified in all the classes of his scheme, and his refinement of its theoretical basis led him to the definition of five Fundamental Categories of facets: PERSONALITY, MATTER, ENERGY, SPACE, and TIME, one of which could incorporate any facet in any subject. The PERSONALITY facet contains (or reflects) the essential character of the subject,

such as KINDS OF BUILDING in the subject BUILDING AND CONSTRUCTION, or EDUCATOR in EDUCATION. The MATTER facet contains (or reflects) the materials of which any member of the PERSONALITY facet might be composed, if the members of the PERSONALITY facet have constituent matter. The ENERGY facet includes any problem, activity, or other intangible conceptual aspect involving the expenditure of energy. It should be noted here that Ranganathan almost always included with the ENERGY facet an AGENT facet, usually referred to as a "second round" of the PERSONALITY facet, since agents were, by and large, things. SPACE and TIME are obvious attributes of any phenomenon.

The Fundamental Categories are an explanation of a pattern rather than a working principle, but they have helped in ordering apparently confusing sets of facets in the development of schemes, and they can be helpful in assigning terms to appropriate facets. Ranganathan used them (or rather he used the shorthand codes by which he referred to them) as an explicit model for the application of each class of his scheme to the classification of documents. For example, in the Colon Classification, 6th edition, in class T EDUCATION he includes facets for:

- EDUCAND, the personality facet [P];
- EDUCATION ACTIVITY, the energy facet [E];
- the AGENT OR METHOD OF THE ACTIVITY, a second "personality" facet [2P];
- the particular KIND OF WORK, a more detailed level [2P2] within that second level that supports the method.

At the beginning of the schedule for EDUCATION he displays the formula:

T [P] : [E][2P], [2P2]

and then lists the details of each facet headed by the appropriate symbol:

Foci in P (= details of Educands)

1	Pre-secondary
13	Pre-school child
15	Elementary
2	Secondary
	(etc.)

Foci in E (= details of Education Activities)

1 Nomenclature
2 Curriculum
3 Teaching technique
(etc.)

Foci in 2P (= details of Academic Discipline)

To be got by (SD) = Subject Device - notation taken from the general classification and included here in parentheses - e.g., ALGEBRA is general class B2

Foci in 2P2 (= details of Method of Activity)
[For 3 Teaching Techniques]

1 Audiovisual
(etc.)

Thus, it is simple to construct a synthesis of notation to describe, for example, AUDIOVISUAL METHODS OF TEACHING ALGEBRA IN ELEMENTARY SCHOOLS:

AUDIOVISUAL METHODS	1 in [2P2]
TEACHING TECHNIQUE	3 in [E]
ALGEBRA	(B2) in [2P]
ELEMENTARY SCHOOLS	15 in [P]

Using the formula T [P] : [E] [2P], [2P2] the notation is assembled as:

T15:3(B2),1

A direct representation of this topic in natural language may display the analysis more clearly:

EDUCATION: ELEMENTARY SCHOOL: TEACHING TECHNIQUES: IN ALGEBRA: AUDIOVISUAL METHODS

This example is included here as indicative of Ranganathan's scheme itself, and as illustrative of his faceted classification theory. The implications of that theory are discussed below.

Ranganathan's ideas were further developed by members of the Classification Research Group (CRG) of Great Britain, chiefly in the 1950's and early 1960's. The CRG was founded in 1950, partly in response to a call for research in classification at the 1948 Royal Society Conference. The Royal Society's intention had been mainly for research into scientific classification, but the CRG was founded by information scientists and librarians, and their interest was in bibliographic and documentary classification. They modified and extended Ranganathan's ideas, and by 1957, had developed a model of faceted classification for special subject areas that remains a classic [NEE57].

The model accepted Ranganathan's Fundamental Categories as a basis for a spectrum of facets rather than as a distinct categorization, arranged in what Ranganathan had already called the order of "decreasing concreteness". Vickery, for example, proposed an extended set of categories of facets [VIC59]: THING (PRODUCT): PART: CONSTITUTENT: PROPERTY: MEASURE: PATIENT: PROCESS/ACTION/OPERATION: AGENT. His classification for soil science recognizes the facets of KINDS OF SOIL: STRUCTURE: CONSTITUENTS: PROPERTIES: PROCESSES (in soil): OPERATIONS (on soil): LABORATORY TECHNIQUES: GENERAL SUBDIVISIONS.

It can thus be seen that classification theory began as a concern with simple structures based on inclusion relation, either of hierarchies of entities, or of hierarchies of processes to which entities were subject; later, it developed into a concern with complex structures involving all kinds of syntactic relation. The use of the term "syntax" in referring to the structure of a classification is no accident. Combinations of terms, combinations by juxtaposition, by association through the use of prepositions or other particles, or in some languages by inflection, are all represented in faceted classification structures.

Problems experienced in classifying bibliographic material of a complex kind (e.g., THE EFFECT OF THE BLACK DEATH (PLAGUE) ON LABOR MOBILITY IN THE ENGLISH WOOL TRADE IN THE FOURTEENTH CENTURY) by classification schemes based on the older, taxonomic model were solved readily by the use of a faceted structure. Not only could such a structure handle descriptions of items as complex as the example above, it could also be

> "a predictive framework capable of accommodating new knowledge without distorting itself, or the knowledge [i.e., descriptions of knowledge], and without the need for complete redesign." [COA73]

6.6 SUMMARY OF FACETED CLASSIFICATION THEORY

This section has described the origins and nature of faceted classification:

- to acquaint the newcomer with the logical progression that has brought faceted classification to its present level;
- to offer some examples of the complexities that faceted classification is designed to handle.

An account of how a faceted classification scheme is constructed and of how it is applied is contained in Sections 9 and 10.

6.7 OTHER APPLICATIONS

Synthetic and faceted classification schemes have been used in all kinds of subject area, from soil science to music, and for many kinds of media, from documents to numeric data in machine-readable form. Mostly they have been used with bibliographic files, but, more than any other kind of index language, classification has been used for nonbibliographic media. The U.S. Patent Office system [NEW57] is a typical example of the use of this kind of syntactical structure to classify descriptions of highly complex physical objects. Classification has also been used in the United Kingdom to handle patent information.

Considerable interest is being expressed currently in the application of faceted classification to photograph and image collections [BAT82]; agencies such as the Central Intelligence Agency realize the very complex nature of the image collection and the varied ways in which it needs to be used.

A very recent example of the use of faceted classification to handle numeric data occurred in the development of a Data Resources Directory (DRD) by the U.S. Department of Energy's Energy Information Administration (EIA) [BAT80a,80b,81]. The DRD's function was to analyze, describe, code, index, locate, and retrieve every complex data element in every EIA data instrument, from forms, through files, to tables and publications. Very complex data elements were reduced to basic data elements, in much the same manner as will be described in Section 9 of this Report.

The scheme in its prototype version was less than forty pages in length, but was able to generate descriptions that represented all components of such complex data elements as GALLONS AND PRICE OF PREMIUM UNLEADED GASOLINE SOLD AT THE PUMP IN A GIVEN MONTH AND REPORTED ON THE 16TH DAY OF THE FOLLOWING MONTH. The scheme was fully worked out for EIA data instruments in the course of the development of the DRD, and is now some 55 pages in length. This illustrates dramatically the economy in physical size of faceted classification schemes, while still handling very complex data elements.

The scheme developed for the DRD was not unusual in its general syntactical structure, so further description is not necessary here. A more detailed account has been included as Appendix A.

SECTION 7

THE NEED FOR FACETED CLASSIFICATION IN THE INFORMATION SYSTEM ENVIRONMENT

The information system environment has two major problem areas which need a better approach to the identification and retrieval of information environment components. The problem areas are:

- o requirements determination;
- o the need to locate information sources.

Faceted classification has high potential for assisting in solving these problems. The advantages of the faceted approach are that:

- o it is methodical;
- o it can be learned readily;
- o it can handle complex data elements efficiently;
- o it has already proven its utility in both the library and data worlds;
- o it is a natural *front-end* to dictionary systems, especially when the dictionary system user does not know exactly what he is looking for.

Although there are many ways in which classification techniques might be used, two particularly significant scenarios illustrate two areas of potential high payback. These scenarios involve:

- o the analyst, during the initial stages of requirements determination;
- o an individual, required by higher management to obtain information on a high priority basis.

In the first scenario, the classification scheme acts first as a source for the user's vocabulary; this has the potential advantage of allowing the analyst to talk the user's language and therefore to understand more readily both the requirements and the impact of the requirements. Once the analyst has determined the information required by the user, a search of the classification structure and dictionary database can yield:

- o the elements which are the information and the containers where that information exists;
- o the elements which can be used to derive the desired information and the containers which contain those elements;

- knowledge that the desired information does not exist within the organization and cannot be generated from data within the organization; therefore, either the information or data which can be used to create the information must be collected from outside the organization.

In the second scenario, a user of this system is searching for information. Through the classification structure, the system can help to identify more clearly what the individual is really asking for. Then the system could:

- identify the containers which contain the information;
- identify the containers which contain elements which can be used as a source to derive the desired information;
- identify through relationships which processes currently manipulate the contents of the containers identified above.

This facility could reduce the current dependence on corporate memory in most organizations, and let the corporate memory reside within the system.

In each of these cases, the use of the system is much like a filter which with increasing granularity can identify precisely the information desired, or with decreasing granularity, can identify sources of information.

To be truly useful, the system should provide a menu-like capability to emulate the classification structure hierarchy. This allows top-down identification of elements and selection of hits. It must also have a synonym capability, because novice users will not know the right words to use. The system must also be able to grow automatically as new requirements are identified. It must also allow automatic restructuring of the classification scheme based on rules defined to it; the resulting restructuring should also update the dictionary and the linkage between the structures where necessary.

Fig. 7:1 illustrates both the logical components of this system, and the relationships and flow between these components. This model illustrates also the need for both databases and functionality not previously discussed. As an aid in understanding the diagram, each concept is explained below:

USER

The individual or automated process which is looking for information. The desired information could be known to a degree which would allow direct retrieval from the dictionary database. In many cases, however,

the precise access name(s) is not known. In this case the user would need to use the classification structure. In either case, it should not be necessary for the user to know both areas; that is the responsibility of the front-end.

FRONT-END

The front-end is software with which the user interacts directly and which uses the classification and dictionary databases to satisfy user requests.

DICTIONARY DATABASE

This database contains data about the components of the information environment of an enterprise. For example, it identifies all elements of data; all forms, reports, and files (containers); and all processes which operate on the containers; it also provides relationships between these various components, e.g., which elements do the containers contain.

CLASSIFICATION DATABASE

This database contains the enterprise words which can be used to identify each of the information environment components by its facets. This is particularly useful for identifying elements and will be used extensively by the front-end to do just that.

CROSS-REFERENCE DATABASE

This database is the common linkage between the other two databases. It is used by the front-end to access the classification database based on data in the dictionary database, or to access the dictionary database based on data in the classification database.

U/R (UPDATE/RETRIEVAL)

This processing function services the access requirements of the front-end when as part of the processing actions on one database require retrieval from or update to one of the other databases.

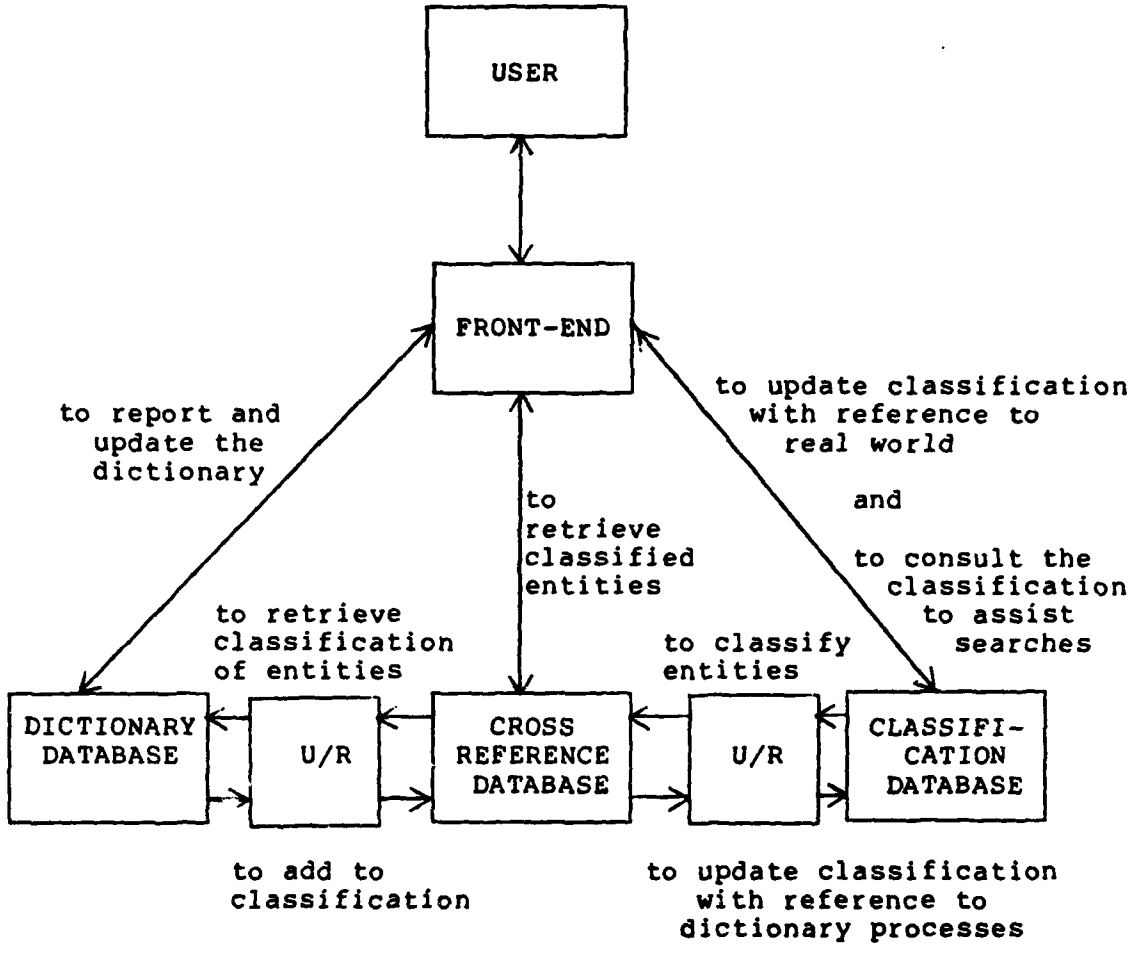


Figure 7:1 Overview of the Use of Faceted Classification as a Front-End to an Information System Using a Data Dictionary

SECTION 8

AN ANALYTICO-SYNTHETIC MODEL OF THE INFORMATION ENVIRONMENT

8.1 INTRODUCTION

As outlined in the previous section, it is the intention of the data classification to support a formal and coded representation or view or a series of representations, of the information environment. It does this through an iterative process of analysis, synthesis, and review already represented in Fig. 7:1 and illustrated slightly differently in general terms in Fig. 8:1. These representations or views can be used to check the information environment for any changes resulting from changes in reporting requirements or the availability of information in the real world of the enterprise.

The purpose of this section is to explain the information environment diagram (Fig. 8:4) that is central to an understanding of the application of the analytico-synthetic and faceted classification structure to the information environment, and then to describe in outline how the classification process works. This broad picture of the classification process will facilitate the detailed explanation of the developmental and application phases of the classification given in the next two sections.

8.2 THE INFORMATION ENVIRONMENT DIAGRAM

The starting point of the development of the information environment diagram was the ANSI/SPARC diagram (Fig. 8:2) representing the levels of representation of data. This was developed to include the classification and a conceptual schema organized by the classification, to support an internal level containing both database and programs (Fig. 8:3). At the same time the concept of iteration was introduced, to allow changes to be made to the database or the programs as changed reporting or other needs had an impact on the system. Finally (Fig. 8:4) the concept of iteration was recognized fully and developed, not only as a simple iteration of the system development life cycle, but also as the iteration of individual stages and at different levels.

8.3 ANALYZING THE INFORMATION ENVIRONMENT

8.3.1 The Data Sample

The first stage of the process illustrated in Fig. 8:4 is an analysis of a sample of data from the information environment, in the form of documents, system documentation, data gathered through interviews with end users about their functions in the

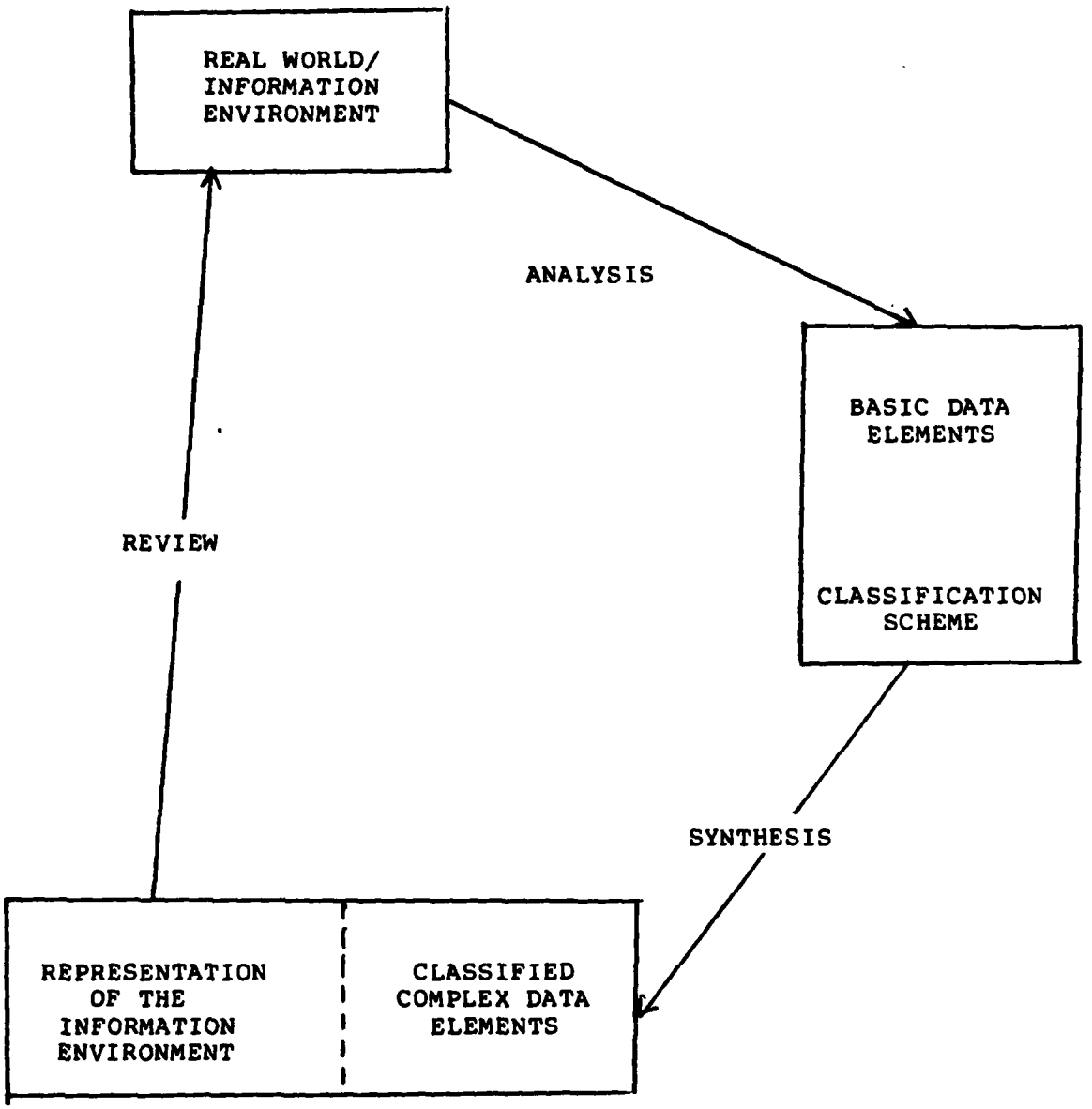


Figure 8:1 Outline of the Relationship of the Information Environment, the Classification, and the Representation of the Information Environment

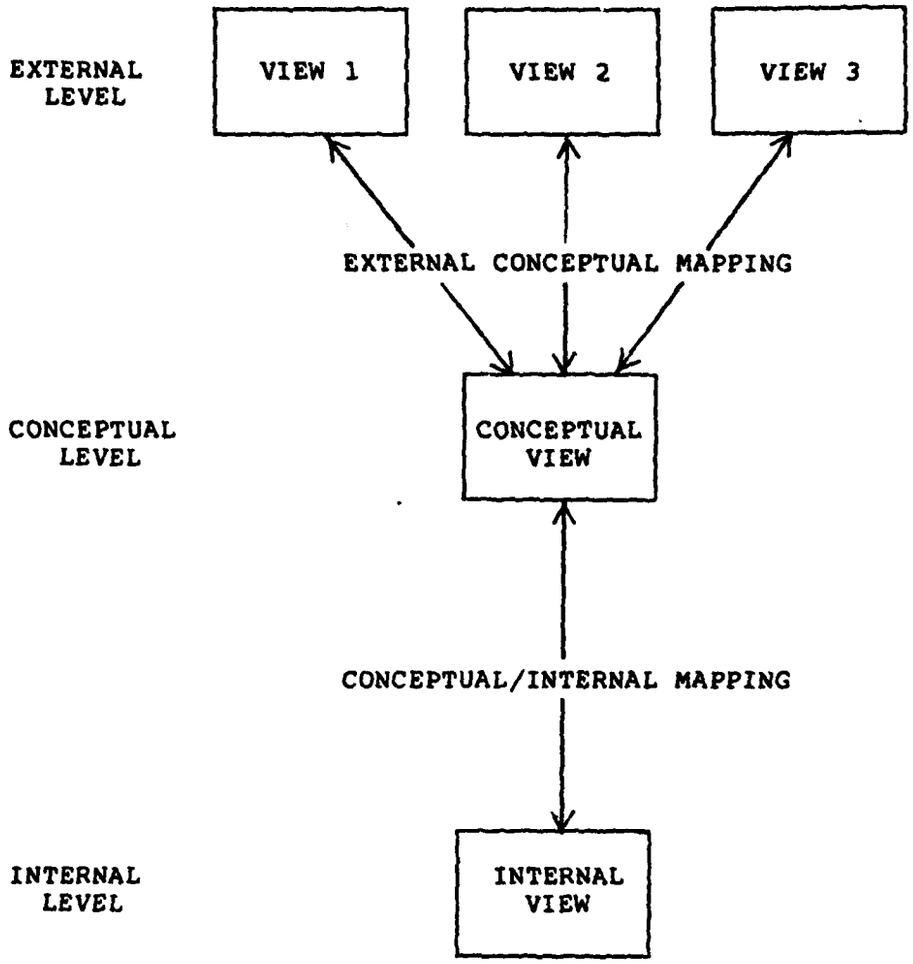


Figure 8:2 After ANSI/SPARC (Bull. of ACM, SIGMOD 7:1)

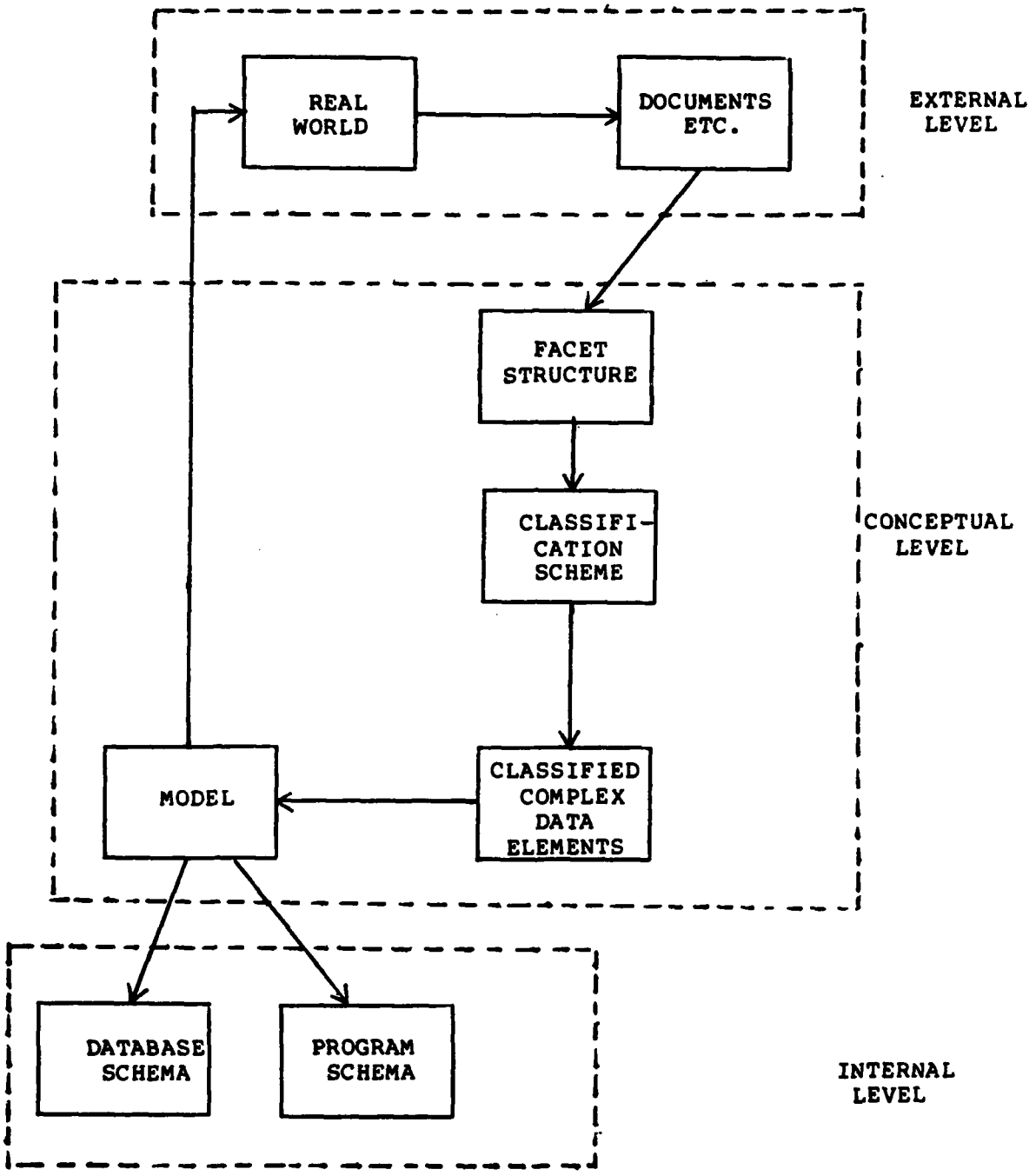


Figure 8:3 Interim Information Environment Diagram

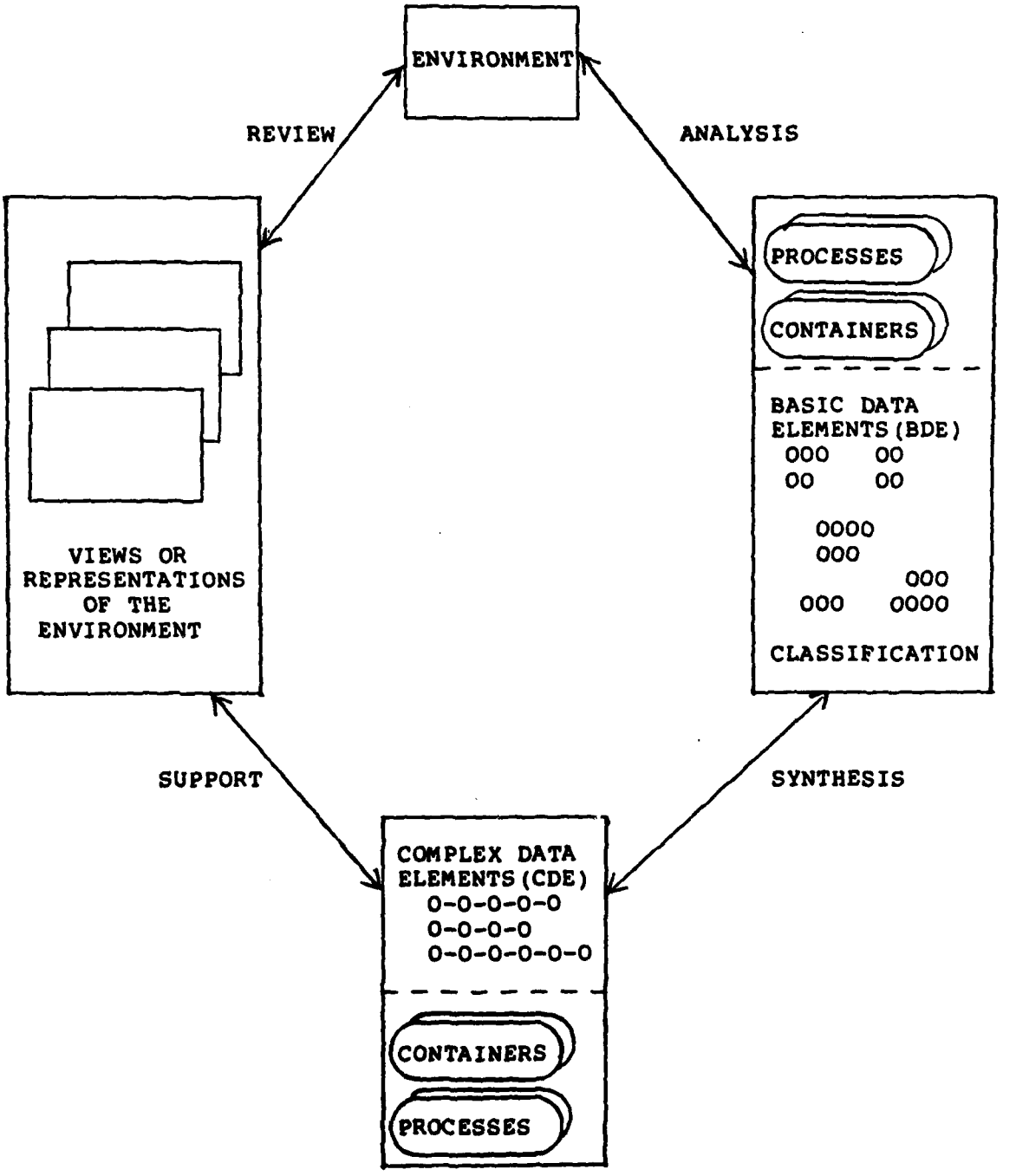


Figure 8:4 The Information Environment

enterprise, etc. For reasons arising from the nature of faceted classification, the sample need not be a very large one to determine the outline of the facet structure, though of course it should be taken from and represent a broad view of the enterprise and its information environment.

The sample is seen as containing a set of raw complex data elements (CDE's) as they appear in the information environment, that is a set of entities each with a number of attributes. Entities are potentially repeatable, i.e., an entity may occur several times, each time with a different set of attributes, and, of course, some attributes may occur elsewhere as entities. Also some entities may appear as members of a series, in which each different entity may have the same set of attributes as other entities in the series.

8.3.2 Analysis of the Sample into Basic Data Elements

The raw CDE's are analyzed into their constituent basic data elements (BDE). For example, the CDE:

> HOURLY READING OF PATIENT'S PULSE RATE AND TEMPERATURE BY ANALOG DEVICE

yields the following BDE's:

> HOURLY
> READING (=MONITORING)
> PATIENT
> PULSE RATE
> TEMPERATURE
> ANALOG DEVICE

These raw BDE's, together with information about the containers from which they were taken, and the information processes in which they are involved, form the basis of box B in Fig. 8:3. The initial process of analysis that decomposes raw CDE's into raw BDE's is an iterative one, continuing until the number of new types of raw BDE's begins to lessen.

8.3.3 Development of the Classification

There is then a further stage in the process of analysis, in which raw BDE's are clustered by common characteristics to begin to develop first the classification structure, and then the classification itself. For example, in the list of raw BDE's shown above, HOURLY obviously belongs to a cluster of terms referring to time, MONITORING, to one of activities, PATIENT, to one of a specific type of people in the hospital environment (DOCTOR would belong to a different cluster), PULSE RATE and TEMPERATURE both belong to one of patient phenomena, and ANALOG DEVICE to a cluster containing equipment.

The recognition of these clusters facilitates the further decomposition of raw CDE's into raw BDE's, and the assignment of raw BDE's to their place in the classification. This part of the process reveals homonyms, synonyms, clerical errors of misspelling, etc., by the conceptual grouping that takes place. The resulting classification is a set of refined BDE's and a set of rules for their use in describing the information environment.

8.3.4 Construct Formalized Complex Data Element Descriptions

The next major process is to construct formalized descriptions of raw CDE's by representing each component BDE by the appropriate refined BDE taken from the classification. For example, if READING is now represented as MONITORING, then MONITORING must be the term used in a refined CDE referring to the kind of situation used in the example above, even if the raw CDE, and the instance existing in the real world, continue to refer to READING. In this way the set of refined CDE's can be checked for redundancy of information and its use in the information environment, and can be used to support a coherent representation of the information environment.

The process of synthesis just described is not usually iterative in itself, but once the classification has reached a stable prototype stage (as described in detail in Sections 9 and 10), it interacts with the process of analysis, as raw CDE's are decomposed into raw BDE's to be matched with the classification. That is, the process of developing a refined CDE begins with the decomposition of a raw CDE into its raw BDE's; these are matched with the classification, and are translated into the appropriate term used as the refined BDE, and the refined BDE's are assembled to make the refined CDE.

Obviously once the classification is stable the process of introducing new BDE's into the scheme changes to one of recognizing which refined BDE is the authoritative term for a raw BDE.

8.4 DEVELOPMENT OF THE REPRESENTATION OF THE INFORMATION ENVIRONMENT

The third process is the development of the representation of the information environment in terms of the refined CDE's. The nature and level of this representation obviously depends on the information environment itself and the needs of management. In general terms, it will have several levels: from a broad picture with only shallow detail, to a level in which full detail is given. At the fully detailed level several interlocking representations will be necessary.

8.5 REVIEW OF THE REPRESENTATION OF THE INFORMATION ENVIRONMENT

The fourth process is the review of the information environment in terms of the representations, to see if any changes in the information environment mean that CDE's, or processes or containers involving them, are no longer accurately represented in the system.

If such changes have occurred, and if anomalies now exist in the system, the whole set of processes is brought into play again. The raw CDE's that now trigger the recognition of the anomalies are analyzed into their constituent raw BDE's; these are matched against the classification; the classification is updated; the raw CDE's are converted to refined CDE's; and the representation of the information environment can again be a true one.

SECTION 9

DEVELOPMENT OF THE FACET STRUCTURE AND THE CLASSIFICATION SCHEME

9.1 INTRODUCTION

The application of faceted classification techniques to the analysis and description of a specific information environment does not differ fundamentally from the classic methodology of faceted classification construction (described in Section 6). Indeed, as can be seen from our description of the experience of the Department of Energy in the prototype development and implementation of the Data Resources Directory (described in Section 6.8 and Appendix A), the task of analyzing a large and complex non-numeric environment is greatly simplified by the use of such techniques. However, such an analysis could be done normally only by an expert in faceted classification, and such an expert would not necessarily know enough about the problems of database design, or about the hardware and software available to the enterprise. Therefore, the present project is intended to describe the design and future development of a methodology and a model for applying facet analysis and faceted classification techniques to any information environment.

There would be two products: a multi-faceted classification of the single-characteristic components of records or complex data elements in the information environment; and a systematic catalog of articulated descriptions of those complex data elements.

The information environment contains three kinds or levels of information:

data
processes on data
containers of data

The data level is the largest and most complex level of information; it includes references to:

- things;
- properties of those things;
- activities, functions or processes involving things;
- tools that support the activities;
- human or institutional participants in the activities;
- indications of locations and time.

Any or all of these may qualify or modify each other, and also may need quantification (e.g., 3% sulfur content). In the sense that the mere presence of an entity in one of these facets provides its value, then all facets are quantified.

In addition to the data, there are the processes by which the data are accepted, modified, and delivered, and the containers or instruments in which data are collected or reside. These processes and containers may be organized by the same classification structure developed for the data, but it is more likely that a modified or even a special classification will be needed.

The model and the methodology involve two levels of organization:

- The first level is the classification structure or scheme itself. This is the set of facets containing the basic data elements (BDE) derived from an analysis of the information environment. The classification and its method of development are described in detail below.

- The second level is the set of complex data elements (CDE) in the information environment. These elements are classified by the scheme, which therefore provides a means to describe and organize them. The method of applying the classification for this purpose is described in Section 10.

These two levels must not be confused. While the first contains and has organized all the basic data elements that can be discerned in the information environment (at least as far as they have been examined) it is not a description of the information environment; it is only the fabric of language that supports that description. To take an example from the world of documentary classification, the first level is the index language, and the second level is the catalog or index that provides an organized view of, and access to, a collection of documents. As noted in Section 8.3.2,

> HOURLY READING OF PATIENT'S PULSE RATE AND TEMPERATURE BY ANALOG DEVICE

is a CDE; PATIENT, PULSE RATE, TEMPERATURE, etc. are all constituent BDE's existing separately in the classification.

In this section, the methodology will be described in neutral terms, that is, as though non-automated. Problems and factors involved in providing an automated version will be discussed in Section 11. An overview diagram of the method is given in Fig. 9:1. The detailed steps in the methodology are now discussed.

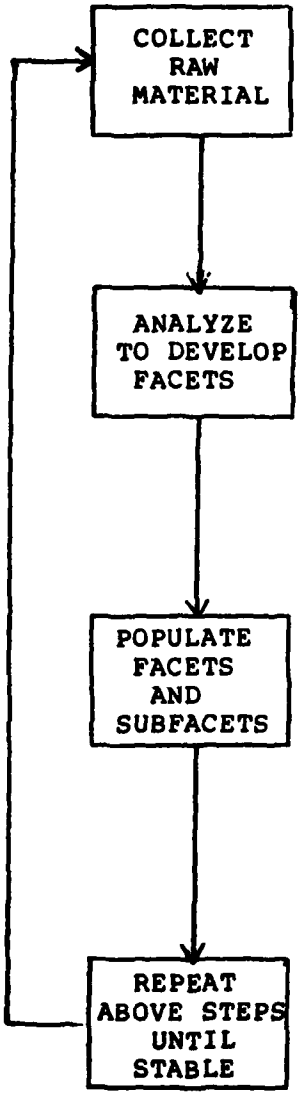


Figure 9:1 An Overview Diagram of the Process

9.2 COLLECT RAW MATERIAL

A variety of raw material exists in an information environment to support its analysis. Typical of these are:

- o forms, reports, and other documents;
- o system and program documentation;
- o models and their documentation;
- o input/output screens;
- o data dictionary contents (if any);
- o end users' activities.

It is one of the fundamental principles of the model and methodology proposed in this Report that it be predictive; therefore in the initial stages of the development of the data classification, only a representative sample of each kind of raw material need be collected.

9.2.1 Size of the Sample

The first four of the above types of raw material are, in one way or another, part of the information environment itself. Forms, reports, and input/output screens support and are in turn supported by the information environment; system documentation supports the processes used in the information environment. All of these, to some measure, contain and/or display complex data elements (CDE), and (in so doing) reveal or can be made to reveal the basic data elements (BDE) needed for the classification. Ultimately, all of the CDE's will need to be classified for representation in the description of the information environment.

The size of the sample should be as small as possible, while remaining large enough to be truly representative. For example, a large enterprise that needs a rich information environment (in the sense of keeping many kinds of logical records) may be represented as a fairly simple enterprise if it is involved in only one general activity, and the range of a classification to describe it is not very great. Thus the classification may be constructed on a fairly small sample set of materials, because there will be frequent recurrence of the same basic data elements, as used in many derived elements.

A good example of such recurrence of BDE's might be found in the coal industry. The coal industry, though large and complex organizationally, is concerned with a limited number of BDE's:

- types, grades and properties of coal;
- kinds of line or seam in which the coal resides;
- recovery and processing activities;
- plant, tools and equipment;
- transportation, storage and delivery activites;
- general and administrative activities;
- locations and time factors, etc.

In each of the above facets (for that is what they are) there may seldom be more than a dozen to twenty BDE's.

On the other hand, a relatively small enterprise may need an information environment so complex that a very broad range of raw material will be needed for its analysis. Indeed, it is likely that a very complex enterprise would need several data classifications, designed to be mutually exclusive in facet types that refer to special areas, but with interlinking or common facet types or common areas; for example, (probably most typically) the administration, accounting, and budget areas.

On evidence so far examined and on consideration of the kinds of material involved, the raw material most commonly found and most useful is the set of forms, reports, etc., used for data collection, modification, and presentation. If a data dictionary already exists (in any form), then it could also provide useful raw material, depending on how well the original analysis of the information environment was carried out. It must be remembered that the objective of the methodology is to allow and encourage more comprehensive, yet simpler analysis than has been possible previously. It is therefore better to develop and use a methodology that is based on primary materials.

9.3 ANALYZE THE RAW MATERIAL TO DEVELOP FACET TYPES

This stage in the methodology has several steps. And just as the general methodology is iterative, so is this stage. The steps are:

- decompose complex data elements taken from the raw material into BDE's;
- assign BDE's to general categories of facets;
- cluster/group BDE's within general categories of facets;
- develop facet types from these clusters;

- o assign BDE's to facet types to test their validity; and
- o develop subfacet types within facets.

Each of these will now be discussed.

9.3.1 Decompose CDE's into BDE's

This step must be carried out carefully and with rigor. Most CDE's have simple and obvious components, as in the example used already:

> HOURLY READING OF PATIENT'S PULSE RATE AND TEMPERATURE BY ANALOG DEVICE

Indeed most CDE's will not be as complex as this example. However, some kinds of raw material can be complex and deceptive; for example, a data collection form about production figures may be laid out as:

DATE	OPERATOR ID	LATHE #	PRODUCT ID	NUMBER PRODUCED	HOURS WORKED

The analysis of this CDE will reveal that it contains two BDE's concerning the machine used (i.e., LATHE #): that it is a lathe, and that it has an ID number. This kind of problem occurs most frequently on printed data collection forms and reports. Fortunately, the later steps in developing the facets and subfacets serve a backstop function, since any complex pseudo-BDE will be revealed when it is found that it cannot be assigned to a single simple array in a facet or subfacet.

Sometimes, though more rarely, a data collection form will contain what are really multiple CDE's on the same line, i.e., where several BDE's are derived from one another as the form is completed. For example:

1	2	3	4	5	6
1" DRILL BITS	IN STOCK ROOM	IN USE	TOTAL INVENTORY (col 2 + col 3)	OBLIGATED	AVAILABLE (col 4 - col 5)

is a complex data element that conceals several constituent complex data elements:

1" DRILLBITS / IN STOCK ROOM
1" DRILLBITS / IN USE
1" DRILLBITS / IN TOTAL INVENTORY
1" DRILLBITS / OBLIGATED
1" DRILLBITS / AVAILABLE

One clue to the occurrence of the phenomenon is the appearance of several BDE's of the same type -- in this case NUMBER. As discussed in Section 10, such a situation is not allowed in explicit classification statements except in very special controlled circumstances. The occurrence of hidden multiple CDE's in a single CDE is a substantial problem at this point in the analysis; however, when the CDE classified descriptions are being assembled in the preparation of the information environment description, this fact is of paramount importance.

In practice this step of decomposition will take place with the support of the techniques described in the next Section.

9.3.2 Assign BDE's to General Categories

The general categories developed for the model and methodology are:

1. THINGS, SUBSTANCES, ENTITIES
2. PARTS OF THINGS
3. SYSTEMS OF THINGS
4. ATTRIBUTES OF THINGS (Properties, Behavior)
5. RELATIONS BETWEEN THINGS
6. OPERATIONS ON THINGS
7. LOCATION
8. TIME

This general list or menu may be expanded and exemplified:

1. THINGS

1.1	Natural	e.g., tree
1.2	Artifacts	e.g., chair
1.3	Mentefacts	e.g., the formula E=MC2

2. PARTS

2.1	Constituents	e.g., wood
2.2	Organs	e.g., legs

3. SYSTEMS e.g., heating/ventilating systems

4. ATTRIBUTES

4.1	Properties	
4.1.1	Structure	e.g., sulfur content
4.1.2	Measure	e.g., tons
4.2	Behavior	e.g., viscosity
5.	RELATIONS	
5.1	Effects	e.g., - does to -
5.2	Reactions	e.g., - combines with -
6.	OPERATION	
6.1	Physical	e.g., welding
6.2	Non-physical	e.g., calculation
7.	LOCATION	
7.1	Specific	e.g., Washington, D.C.
7.2	Qualifier	e.g., southwest
8.	TIME	
8.1	Duration	e.g., year
8.2	Periodicity	e.g., annually
8.3	Qualifier	e.g., previous

As BDE's are recognized in the examination of the raw material, they may be assigned to one of the general categories. This assignment will not automatically develop facet types or facets. Instead it will limit by function the range of BDE's to be analyzed into facets. In other words, the BDE's assigned to the THINGS (Artifacts) category will be a comparatively small subset of the total set of BDE's in the information environment. This will simplify the next step.

9.3.3 Cluster within Categories

Each general category in turn is examined to discern clusters of BDE's that share common characteristics. This is an intellectual process that may be supported by some part of the automated process, but it is in essence a recognition of intrinsic similarities. For example, the category OPERATIONS (physical) may contain BDE's like BENDING, SHEARING, SHIPPING, PAINTING, and STORING. An examination of these terms might yield these clusters: [construction] BENDING, SHEARING, PAINTING; [transportation] SHIPPING; [storage] STORING.

This process is iterative in itself, as some BDE's will not fit into clusters already formed, and by their anomaly, will cause the clusters to be reformed. As the clusters become stable, the recognition of their characteristics allows us to develop the facet types, which form the operationalized level below the general categories.

9.3.4 Develop Facet Types

When the clusters of BDE's seem stable, each cluster is examined to see what characteristic it reflects. This enables its facet type to be determined. Some characteristics are easy to recognize; others are more difficult.

The example used in the previous section, BENDING, SHEARING, PAINTING, reveals terms that share the characteristic of a general process of construction, and at this stage of developing facet types such a general characterization might be sufficient. Already, however, it might be necessary to recognize a further sub-characterization - with BENDING and SHEARING in one cluster, and PAINTING in another - because they belong to different stages of the process of construction.

Testing the facet types, both for their general validity and for the appropriate level of subdivision, belongs to the next step.

9.3.5 Populate the Facet Types

It is now necessary to examine all BDE's so far collected from the decomposition of CDE's to check:

- that they all belong to one facet type;
- that in no facet (as the populated facet type has become) are there any BDE's that display a different characteristic; and
- that in no facet are there terms that, though they may be mutually exclusive, still may be combinable.

If the last case does not hold, then a further step must be taken.

9.3.6 Develop Levels or Sub-facets

If mutually exclusive BDE's may be usefully combined to describe more complex items, it is necessary to decompose the facet further. There are two conditions that may obtain:

- There may be two or more characteristics that produce clusters whose members may be combined. For example, in TIME there may be a need to develop one list of CHRONOLOGICAL DIVISIONS (e.g., DATES) and also another list of OTHER TIME ASPECTS, like PERIODICITY (e.g., WEEKLY), to allow a statement like WEEKLY PRODUCTION IN 1982. These lists constitute subfacets.

- The arrays of increasing subdivision in a facet may constitute sets of combinable terms, i.e., where any term from a lower array may be used to modify any term from a higher. For example, a MACHINE has PARTS (e.g., BED, POWER UNIT) that themselves have basic PARTS (e.g., PLATE, BOLT, BRACKET). These constitute levels that are regarded much the same as subfacets.

Fig. 9:2 illustrates these two conditions.

There is another situation in which subfacets may exist that does not involve (and may even preclude) the combination of constituent terms. This occurs when a facet may be divided into parallel hierarchies that contain the same terms but organized in a different way.

For example, PETROLEUM PRODUCTS may be characterized by their position in the fractionating column. However, some products or groups of products are not (and cannot be) so characterized, e.g., AVIATION FUEL. In such a case, two parallel subfacets must be created as alternatives to each other: in this case PETROLEUM-PRODUCTS-BY-CONSTITUENT and PETROLEUM-PRODUCTS-BY-USE. The same terms may appear in both subfacets (e.g., MOTOR GASOLINE) thus creating an apparent homonym situation, and therefore the term must clearly indicate which subfacet should be used.

This discussion of subfacets and levels has been included here for completeness. As will be seen, complexities of this kind (common in the bibliographic world which gave rise to them) may not be encountered sufficiently frequently or with sufficient severity to warrant their attention in developing a data classification. Detailed examination of this problem would be appropriate for the experimental implementation phase of the project.

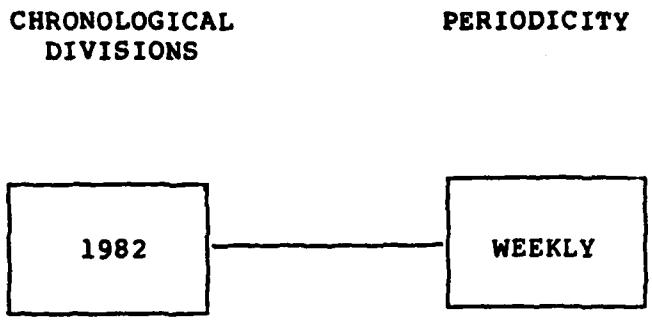
9.4 CONSTRUCT THE FACETS, SUBFACETS, ETC.

This stage of the development of a data classification was already started in the previous stage described in Section 9.3 above. In that stage, the assignment of BDE's to facet types to test them has already begun to populate the facet types. What remains is to populate them further in order to develop the scheme to the fullest extent possible at this early stage.

9.4.1 Populate the Facets

Once the facet types have been determined it is a comparatively simple task to look for other terms that share the characteristic of the facet type. That is, once a facet type MACHINE has been approved, it is a simple matter either to look for an inventory of machines already kept by the enterprise, or to be ready to allocate to this facet any reference to a machine when encountered in collecting and reviewing more data about the

SUBFACETS:



LEVELS:

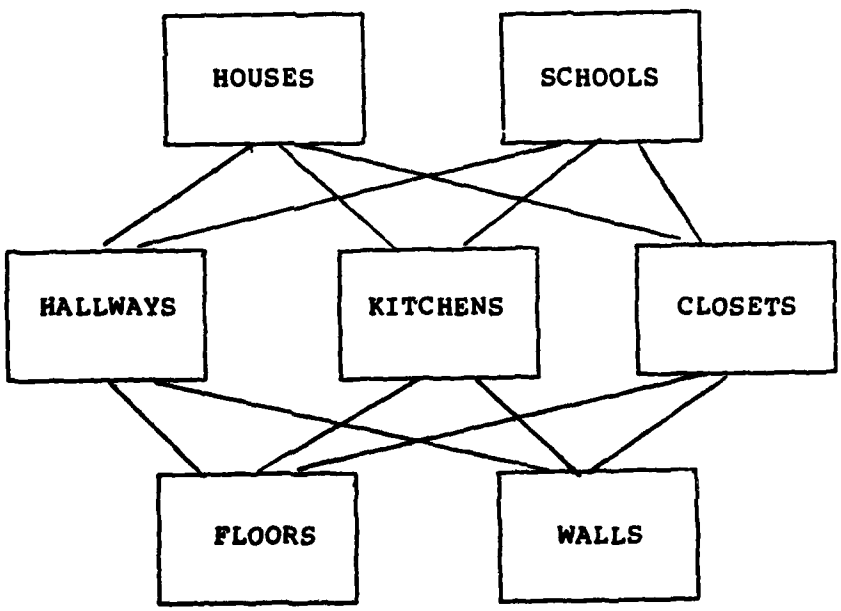


Figure 9:2 Subfacets and Levels

enterprise. This shows that there is a predictive feature incorporated in the faceted classification structure.

The iterative process of collecting and analyzing the data and populating the classification scheme requires several passes through the system. In this iterative process the stages of analysis and construction overlap and even blend; this does not change their characteristic and different natures. In the early period of constructing the data classification, analysis is paramount, the clusters will change and reshape, and the facet types will not be heavily populated. In the middle period, as the facet types stabilize, the population will increase rapidly, until the last (or rather a later) period, when the growth curve begins to flatten.

At this point, the data classification is ready to be used to classify and organize CDE's, though of course it will never really cease to be populated further when new raw CDE's are encountered requiring new BDE's (i.e., BDE's not yet specified in the scheme).

9.4.2 Populate the Subfacets

This process is exactly the same as the process of populating the facets, with the one additional problem of ensuring that foci (the individual terms) are included effectively in only one subfacet.

9.4.3 Check the Arrays

This is a relatively minor task in classification development. As a facet is developed, its constituent terms will not all be at the same level of division. It is common (even inevitable) to find small, simple hierarchies of generic/specific relationships. Each level of coordinate terms is an array that must contain terms that are both mutually exclusive and collectively exhaustive, i.e., they reflect only one common characteristic, and the "sum" of all the terms is equal to the range of meaning of the facet type. However, it is possible that in a constantly developing data classification, collective exhaustivity is not a major factor, and therefore not a major problem.

9.4.4 Check Synonyms, Misspellings, and Homonyms

This is a most important task and a special feature of the data classification. In the development of data dictionaries, for example, there are few more troublesome problems. Facet analysis can simplify this in several ways:

- The collection of the BDE's into facets, subfacets, and arrays reduces the range of objects that must be scanned for synonyms. Indeed, as the facets develop, synonyms should be brought more closely together, since they represent the same or similar concepts.

- As CDE's are analyzed into BDE's, a method can be used to alert the classification developer of the possibility of a synonym. This method is to restrict the description of a CDE to one term in each facet that applies to the CDE. If the CDE reflects a facet more than once, then two CDE descriptions must be formed. For example, the Social Security Card (U.S. Government Form 0A702) contains two BDE's: the NAME and the SSN. Since both NAME and the SSN would belong in a facet of PERSONS, two descriptions would be made up:

  - NAME [PERSONS]: 0A702
  - SSN [PERSONS]: 0A702

  An automated system could flag the presence of two values in the same facet for the same container; the definer would thus be alerted to check whether the two values are synonymous or not.

- Homonyms are not always easily picked up by the faceted display of the data classification. If they occur in the same facet, the limited range of terms available for scanning improves the chances of detection. However, this situation is unlikely if they occur in different facets. Then an alphabetical listing of terms, together with the indication of their facets, will reveal them.

9.5 ORDER OF FACETS AND SUBFACETS

The order of facets and subfacets is an important one in faceted classification when used in a manual mode, because the resulting strings will be used as visual sorting and filing codes. The most significant facet will always appear first in a string, followed by the next significant, and so on, and there are fairly elaborate rules for the establishment of such a facet order.

In the present situation, the data classification will not be used to organize the collections of CDE's as though they were documents in a file, and therefore these rules are not relevant. However, a consistent order of assembly of the facets, or rather of the BDE's from those facets, is desirable both in machine-readable and visual forms, for the purposes of checking and comparison. That consistent order should be decided by the analyst as the facets are developed, so that it may be used in the application of the data classification described in the next Section.

9.6 NOTATION FOR THE FACET STRUCTURE

One of the best ways to control the conceptual map at the manual or visual level of the faceted data classification, is a notation or set of codes assigned to the facets, subfacets, and constituent isolates. Of course in a machine-readable form, such a notation is not needed for purely organizational purposes, though it may have some utility in special circumstances (depending on the software available) for searching or comparing.

If notation is used, its application to the growing data classification should follow several basic rules:

- The notation should be alphabetic, to allow a large notational base.
- It should be fully expressive, i.e., the notation should be extended by one character for every successive step of division.
- The letter A should never be used to end a notational code, to allow for the insertion of new BDE's at the beginning of an array.
- In an automated environment the notation may be applied mechanically by the system to the growing data classification. This means that the notation assigned to a given BDE may be different from one version of the scheme to another, but since the primary function of the notation is to control the scheme, and not (as in most bibliographic schemes) to code the CDE's, then variation of the notation over time is acceptable.

SECTION 10

APPLICATION OF THE CLASSIFICATION TO THE INFORMATION ENVIRONMENT DATA

The application of the faceted classification to the information environment is in one sense an extension of the work of developing the scheme. Raw complex data elements (CDE) are analyzed in terms of the facet structure, and the controlled language components are then subsequently assembled into the refined CDE description.

However, in another sense the work is fundamentally different. Once the application stage is reached, the classification scheme is in at least the prototype stage; therefore, there is no on-going reorganization of the classification structure in response to new information derived from the decomposition of the raw CDE's. This is not to say that reorganization and further development can no longer take place, but it is not done in the same way as in the developmental stage. Any proposed changes must now be considered as candidate changes only, and the classification scheme will be reviewed for updating at definite and regular intervals.

Simple extensions of the classification scheme to accommodate new basic data element (BDE) names that belong without doubt or confusion to existing arrays within existing facets can, of course, be included in the same on-going manner as in the developmental stage.

The application stage has another fundamental difference from the developmental stage. Since the classification structure is now relatively stable, its application must be according to some general rules, which existed as principles in the developmental stage, but which now emerge as rules for application in the specific information environment.

For example, if a raw CDE contains two instances from the same facet, it is natural to ask if the two instances belong to different, mutually exclusive subfacets, or if they really belong in the same subfacet and even array. For example, a raw CDE

THOUSANDS OF TONS OF BITUMINOUS COAL AND LIGNITE

might have one of two quite different meanings:

THOUSANDS OF TONS OF BITUMINOUS COAL AND LIGNITE
(an aggregated total)

THOUSANDS OF TONS OF BITUMINOUS COAL AND [OF] LIGNITE

In the second case the raw CDE actually represents two raw CDE's. In Boolean terms the first case is an ANDed relationship, and in the second an ORed relationship.

In the developmental stage such a determination primarily assists the classificationist in perceiving what BDE's need specific expression and what may be left as a term representing an aggregation. In the above case, the phrase BITUMINOUS COAL AND LIGNITE might represent an aggregation that is never broken down into its specific components. If, as is much more likely, the aggregation is broken down into the components BITUMINOUS COAL and LIGNITE, then the classificationist should record a raw CDE (and source) for each component.

In the application stage such a determination and such a recording is mandatory. Except in very special circumstances and with appropriate signals (by the use of explicit operators), a refined CDE must not contain two BDE's from the same array in the classification, and must never contain them in an ORed relationship.

In practical terms, application of the classification to raw CDE's in order to create refined CDE's requires a number of stages similar in nature but not in detail to the stages described in the developmental phase:

- analysis of the raw CDE into its BDE's;
- checking that all facets have been used to analyze the raw CDE (some facets may be empty with respect to a given raw CDE);
- tracing the most specific level of the hierarchy in each facet to find the most specific BDE and its approved language;
- assembling refined BDE's into refined CDE's according to the general syntactical order of the classification;
- checking if operators are needed to override the general syntactical order with its implied semantic context, in order to ensure that the refined CDE represents the raw CDE fully and correctly.

There are several major exceptions to the prohibition of two BDE's from the same array in the same refined CDE. Each of them requires the use of an explicit operator to indicate the kind of overriding consideration:

- AND: the Boolean AND (logical conjunction) -- used for aggregation, e.g.:

  THOUSANDS OF TONS OF BITUMINOUS COAL AND LIGNITE

  COST OF SHIPPING AND STORING

  [NUMBERS OF] KNIVES AND FORKS AND SPOONS IN THE CAFETERIA

- EXCEPT -- used to refer to a class of which only one subclass is missing. It avoids the creation of many ORed refined CDE's from one raw CDE, e.g.:

  COAL EXCEPT ANTHRACITE

- OR: the Boolean OR (logical disjunction) -- used in special circumstances in particular systems to trigger the creation of two or more refined CDE's. This use of OR is at the system level, rather than the intellectual level, and is only to avoid the repetitious work of entering several refined CDE's based on the decomposition of one raw CDE.

- FROM, TO, IN: English language prepositions referring to location -- used to allow multiple locations in a single refined CDE, e.g.:

  LOADING DOCK TO STORAGE AREA

- OUT OF, INTO -- used to link a source material and a product if the two terms belong in the same array or subfacet. The choice of operator depends on the orientation of the refined CDE, e.g.:

  LENGTH OF FINISHED PLANKS OUT OF RAW PLANKS

  LENGTH OF RAW PLANKS INTO FINISHED PLANKS

- PER -- used to link two terms from the same array or subfacet in a quantitative, usually a measuring, facet, e.g.:

  GALLONS PER HOUR

  FEET PER SECOND PER SECOND

- OTHER -- used to indicate that the class term used is incomplete. There are many situations in which this operator is virtually useless.

The above list of operators is indicative, not definitive. In any specific information environment the precise function, description, and rules for application of any operator may need to be modified, and new operators may need to be identified.

SECTION 11

DEVELOPMENT OF USER FRIENDLY AIDS IN CLASSIFICATION DEVELOPMENT

11.1 INTRODUCTION

The sequence of steps in the developmental and application phases of the classification were described in Sections 9 and 10 in a neutral mode. That is, the description did not refer specifically to a manual- or a computer-assisted activity.

Clearly at the heart of both the developmental and the application phases lie two related but distinct intellectual operations. In the developmental phase, it is the analysis of the raw complex data elements (CDE) into raw basic data elements (BDE), and the organization of the raw BDE's into the facet structure, converting them to refined BDE's. In the application phase, it is the analysis of raw CDE's into raw BDE's, and their translation into an assembly of refined BDE's to make refined CDE's.

The system will never be able to take on the work of intellectual analysis, but most other steps in the phases may be supported by the computer, and even in the intellectual operations, it is possible to offer computer-supported guidance.

The general form of a user-oriented developmental program includes a manual and a series of menu-driven prompt programs to guide the user through the sequence of steps or stages required to build the classification. Each stage takes the user to the next more specific stage. At each stage, there is an opportunity for the user to repeat the step or to return to an earlier stage; this supports the iterative nature of the process.

11.2 COLLECT RAW MATERIAL

This is a physical stage in the process and will be supported by the system only to the extent:

- of prompting the user with a list of types of material usually available;
- of prompting the user to record in the system the names, descriptions, and reference codes of the raw material to be used.

An additional feature might be included in this stage: to have the system recommend the number of the sample items of raw material. This number would be based on a formula involving data like the roughly estimated number of data instruments in the information environment, the number of personnel, the number of products (if relevant), and the number of units in the

enterprise. Such a formula would be highly complex, and would require testing in a practical demonstration.

11.3 ANALYZE THE RAW MATERIAL

This stage in the methodology has several steps. Just as the general methodology is iterative, so is this stage within itself. The computer-supported system allows the user to remain with a step until all necessary work is completed and it is possible to move on to the next step.

11.3.1 Decompose Raw Complex Data Elements (CDE's) Taken From the Raw Material

Two modes of operation will be possible: the entry of raw CDE statements by data entry personnel *en bloc*, for later decomposition by the user; or the entry of raw CDE's one at a time by the user, for immediate decomposition. In either case, the system should accept strings of characters (phrases descriptive of the raw CDE's) together with a source reference code (SRC) for the raw material item from which each raw CDE is drawn, and in the case of entry *en bloc* should store the strings in an accessible file.

The system will display the raw CDE's in turn (on retrieval or on one-at-a-time entry) and ask the user if any part or parts of the raw CDE can be assigned to the first of a list of general categories of facets. Block markers, or cursor position, will be used to indicate the words to be so assigned. An additional feature will invite direct keyboard entry to augment or replace the words in the raw CDE. The system will then store the indicated raw BDE together with the reference code for the source.

The list of general categories of facet will be as described in Section 9.3.2 of this Report:

THINGS, ENTITIES

PARTS OF THINGS

SYSTEMS OF THINGS

ATTRIBUTES OF THINGS

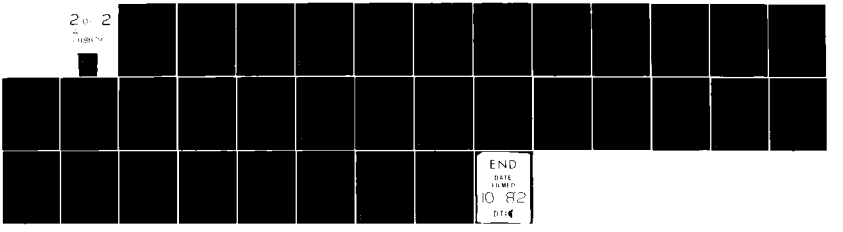
RELATIONS BETWEEN THINGS

OPERATIONS ON THINGS

LOCATION

TIME

AD-A118 879 ALPHA OMEGA GROUP INC SILVER SPRING MD F/G 5/2
DEVELOPMENT OF A USER-ORIENTED DATA CLASSIFICATION FOR INFORMAT--ETC(U)
JUN 82
UNCLASSIFIED AOGR2-ONR-1 N00014-82-C-0129
NL



The expanded list also displayed in Section 9.3.2 may be used when more specificity is desired, e.g.:

THINGS, ENTITIES

Natural
Artifacts
Mentefacts
(etc.)

The system will continue to ask the user for more decomposed elements to match each general category in turn until the raw CDE is exhausted, or until the user indicates the retrieval or entry of the next raw CDE.

11.3.2 Cluster Raw BDE's Within General Categories of Facets

The system, at any time, will display or print out an unordered list of raw BDE's in any requested category. The user may request or suppress the display of the source reference code for the source of the BDE. The user is thus able to examine the terms in the list for groups with common characteristics. The system will invite the user to indicate the terms to be assigned to a group, and also to indicate when the group is complete, at least for the time being. For example, if the general category ENTITIES, Artifacts contains:

SCHOOLS
BUILDINGS
SHIPS
VEHICLES
TRUCKS
HOSPITALS

the user will indicate a decision to assign SCHOOLS, BUILDINGS, and HOSPITALS to a common characteristic group. The system will record that decision, will retrieve and display these terms as now belonging to the group. The system will also ask the user to enter the common characteristic of the group; this term will serve as the facet name.

The system will then prompt the user to indicate a second group, e.g., VEHICLES, TRUCKS, and perhaps SHIPS. If the user realizes that a term assigned to the first cluster should be re-assigned to the second cluster, he can so re-assign it, and the system will record the change.

At any time the user may request the display or printout of the group of terms, with or without the reference code for the source, by calling for any single term. Also the user may request a display or printout of all terms not so far assigned to groups.

As the user proceeds with this analysis and assignment, the groups will become more and more stable. As a group becomes stable, the system will allow the user to proceed to the next step, while still allowing at some future time a return to this step.

11.3.3 Develop Subfacet Types

The system will allow the user at any time to leave the assignment of raw BDE's to groups, and pass to the further examination of a single group now stable enough to be called a facet. This further examination is similar to the examination conducted in the previous step: the facet and its terms are displayed or printed out, and the user may then indicate membership of terms in subfacets within the facet.

The nature and structure of subfacets is described in detail in Section 9.3.6. Subfacets are mutually exclusive subclusters occurring within facets; the terms from any subfacet may be combined with terms from any other subfacet to form complex descriptions, e.g., the facet BUILDINGS may be divided into subfacets of SIZE, METHOD OF CONSTRUCTION, and PURPOSE, to allow a description of the complex data element:

SINGLE-STORY PREFABRICATED HOSPITALS

SIZE, METHOD OF CONSTRUCTION, and PURPOSE are the characteristics which serve as the names of these subfacets within the facet BUILDINGS.

11.4 CONSTRUCT THE FACETS, SUBFACETS, ETC.

The system will continue to accept raw CDE's, either en bloc for later analysis, or by individual entry for immediate analysis. In this stage, the user is presented with a display of all the BDE's collected into a facet and is prompted to construct a generic-specific hierarchy. Because of the preceding step of clustering, there should be not too many BDE's within each facet or subfacet to be scanned at a time.

There are two methods of constructing a hierarchy. Both will be available in the system.

The first method allows the user the top-down approach of first naming those BDE's that have no superordinate level. They are the most general level of the hierarchy; they constitute the most general array. For example, the user may decide, for the facet containing:

SCHOOLS
BUILDINGS
HOSPITALS
JUNIOR HIGH SCHOOLS
SENIOR HIGH SCHOOLS

that the highest level has only BUILDINGS. The system will then display the term BUILDINGS, and ask what terms constitute its next level or array. If the user indicates SCHOOLS and HOSPITALS the system will record the generic-specific relationship between BUILDINGS and SCHOOLS and BUILDINGS and HOSPITALS, and will also record the coordinate (sibling) relationship SCHOOLS and HOSPITALS. The system will then display the first term SCHOOLS and ask for its next level or array. If the user indicates JUNIOR HIGH SCHOOLS and SENIOR HIGH SCHOOLS, the system will again record the relationships, and again display the first term JUNIOR HIGH SCHOOLS with an invitation to enter the next level or array.

This downward process of division continues until the most specific level is reached in the chain under consideration. The system will then return to the next term not yet treated in the next higher array.

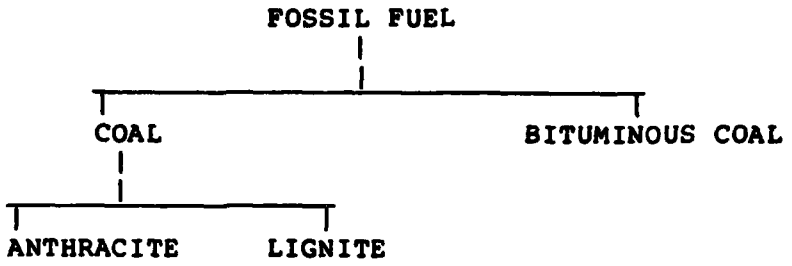
The second method is more flexible. The user may assign a generic-specific relationship between any two terms in the facet, provided he is sure that the two terms are immediately super- and sub-ordinate in the chain of division. The system will store the relationships for the constituent terms, and display the growing chains and arrays on request. This allows the user to correct anomalies formed by the indication of generic-specific relationships that are not close enough. For example, if terms like:

FOSSIL FUEL
COAL
ANTHRACITE
BITUMINOUS COAL
LIGNITE

were paired as:

FOSSIL FUEL - COAL;
COAL - ANTHRACITE;
COAL - LIGNITE;
FOSSIL FUEL - BITUMINOUS COAL

the display would read:



and the obvious anomaly could be seen and corrected.

The first method is surer, but is possible usually only for fairly small facets with clearly apparent hierarchies. The second method is more feasible for large facets, but carries with it the risk of the confusion of levels of the hain.

11.5 CHECK SYNONYMS, MISSPELLINGS, AND HOMONYMS

As the facets develop as more and more highly structured conceptual groupings, possible synonyms will be brought more closely together in the visual display, since they represent the same or similar concepts. Thus any display or printout of a facet will reveal variant expressions of a similar concept, whether caused by the use of synonyms, alternate spellings, or misspellings.

The system can also alert the user to the possibility of a synonym automatically in the application stage, if the description of a CDE is limited to one term in each facet that applies to the CDE. Thus, if a CDE reflects a facet more than once, then two CDE descriptions must be formed. For example, the Social Security Card (U.S. Government Form OA702) contains two BDE's: the NAME and the SSN. Since both NAME and SSN belong in a facet descriptive of persons (and since only one descriptor from one facet may appear in any one description), two separate descriptions would be established:

NAME [PERSONS]: OA702
SSN [PERSONS]: OA702

The system will flag the presence of two values from the same facet for the same container or source reference code and ask if these are synonyms. In this case, of course, they are not; indeed, they would lead to the establishment of two subfacets. The mechanism therefore serves the dual purpose of checking for synonyms and for logical errors.

An alphabetical index of the terms listed in all facets will reveal homonyms and even near homonyms and indicate the facet in which they reside. This will allow the dissociation of homonyms or near homonyms that function differently as verb or object - with a consequent explicit indication in the index language.

11.6 THE APPLICATION OF THE DATA CLASSIFICATION

The system will support the application of the data classification scheme in order to form the list of refined complex data elements (CDE) to describe the information environment in two ways:

11.6.1 Analyzing Raw CDE's

The system will prompt the user to analyze raw CDE's in much the same way as in the developmental phase described in Section 11.3 above with one difference: the names of the facets and constituent subfacets that make up the developed classification for any given enterprise will be used to prompt the user, rather than the names of general categories. For example, the classification of energy data developed for the EIA contains, among others, the facets:

ENERGY SOURCE
ENERGY SOURCE PROPERTY
FUNCTION/ACTIVITY

Therefore, the user would be prompted first for any BDE component of the CDE that might belong to the ENERGY SOURCE facet. After that facet was dealt with satisfactorily, as described below in 11.6.2, the system would ask for any BDE that might belong to the ENERGY SOURCE PROPERTY facet, and so on.

11.6.2 Translating the BDE's into the Language of the Classification

When the user responds to a prompt for a given facet, the system will offer a menu of subfacets, or the terms in the principal array, and will allow the user to select, through a series of menus, refined BDE's at the appropriate level of specificity in the classification. The system will store the acceptable terms until all facets have been examined, and will then assemble or flag the refined BDE's in a consistent order, taking account of operators for overriding special relationships among the BDE's.

SECTION 12

IMPLEMENTATION OF THE SYSTEM AND THE USE OF THE NEW METHODOLOGY

The current work has demonstrated that state-of-the-art requirements analysis methodologies in general lack a systematic way of gathering and organizing data from primary sources. Those methodologies that address this area, and reviews of their efforts in this area, indicate the need to develop a systematic method of data collection and organization.

Faceted classification has a long and well-documented history of handling very complex data elements in the bibliographic field, and more recently of handling nonbibliographic data. Further, it has the ability to combine Boolean searching, with Aristotelean-based, hierarchical scanning.

This Report has described in detail a method for using faceted classification to organize and describe the complex data elements in the information environment of an enterprise. It has also described how the method could be supported by an automated development, application, and maintenance system. It has further described the faceted classification and the description of the information environment in terms of a front-end to a database and data dictionary system. The use of such a systematic method of collecting and organizing data will radically affect subsequent stages of system development, management, and use.

The use of the new methodology and implementation of the automated system should be demonstrated and tested on a limited database of sufficient size. The developmental effort suggested for the demonstration of the efficacy of the system described in this Report involves developing a database of some thousands of records, preferably from a real agency or from a unit of an agency, for example, the accounting and procurement database of a significant agency or enterprise. The unit or the collection of records should be selected to represent as diverse a collection as possible to test the efficacy of the system.

The system would provide the following features and products in machine-readable form (with printed listings when appropriate):

- a classification of all basic data elements used in the information environment;
- accurate descriptions of all complex data elements used in the information environment;
- a complete catalog of all complex data elements, together with their sources and locations;
- a locating and tracking system of all basic and complex data elements;

- multiple level retrieval of all data element descriptions, by any facet, and of course, the display or printout of any set or subset of those descriptions by any key in the classified description;

- a user friendly front-end to the system, involving the classification and a data dictionary system.

The implementation of the system would also develop a prototype of the computer-supported classification methodology for use with other environments, as well as full documentation.

The system would offer a far more flexible and powerful management and planning tool than is commonly found in practice. It would offer a significantly faster development time, or conversely, a more sophisticated development mechanism, and it would allow more timely updates in response to changed requirements.

The level of effort recommended for this implementation and demonstration is of the order of three man years over a period of eighteen months, covering the following activities:

- development of the classification;

- application of the classification to produce the description of the environment;

- development of the database linking classification, database, and data dictionary;

- development of the prototype classification generator;

- system startup;

- testing;

- documentation production.

The work requires expert help in the following areas:

- faceted classification;

- data dictionary systems;

- database management systems;

- software configuration management;

as well as junior data processing support.

The estimated cost of such an effort would be approximately $400,000.

REFERENCES

[ANS75] ANSI/X3/SPARC Study Group Interim Report: Data Base Management Systems, ACM SIGMOD FDT 7, 2, 1975.

[AUS69] AUSTIN, D., "Prospects for a New General Classification." Journal of Librarianship, 1(3), July 1969.

[BAL76] BALKOVICH, E. and G. Engleberg, "Research Toward a Technology to Support the Specification of Data Processing System Performance Requirements." Proceedings Second International Software Engineering Conference, October 1976.

[BAT80a] BATTY, C. D., "Feasibility and Prototype Development of a Faceted Classification of Energy Data for the Energy Information Administration's Data Resources Directory." Paper presented at the American Society for Information Science Mid-Year Conference, May 1980, Pittsburgh. Published by the U.S. Department of Energy, July 1980, as NEIS Memorandum 80-22.

[BAT80b] BATTY, C. D., "The NEIS Classification Scheme." U.S. Department of Energy, October 1980. NEIS Memorandum 80-10.

[BAT81] BATTY, C. D. and Irene Travis, "A Classification-based Information Retrieval System for Federal Energy Data." In Using Information: Papers Contributed to the Tenth Mid-Year Meeting of the American Society for Information Science, Durango, May 1981. ASIS, 1981. Part One: p. 1; Part Two: pp. 2-11.

[BAT82] BATTY, C. D. and Patricia Stevens, "Automated Retrieval Systems for Photo-Image Collections: Problems and a Solution." To be presented at the 45th Annual Meeting of the American Society for Information Science, October 17-21, 1982, Columbus, Ohio.

[BEL76] BELFORD, P. C., A. F. Bond, D. G. Henderson, and L. S. Sellers, "Specifications a Key to Effective Software Development." Proceedings Second International Software Engineering Conference, 1976.

[BLI10] BLISS, Henry E. "A Modern Classification for Libraries ... " Library Journal, 35, 1910, 350-358.

[BRA75] BRATMAN, H. and T. Court, "The Software Factory." Computer, May 1975.

[BRO06] BROWN, James D. Subject Classification. London: Grafton, 1906.

[BUB75] BUBENKO, J. A., "IAM: An Inferential Abstract Modeling Approach to Design of Conceptual Schema." ACM SIGMOD International Conference on Management of Data. Toronto, August 1977.

[BUB80] BUBENKO, J. A., Jr., "On Concepts and Strategies for Requirements and Information Analysis." SYSLAB Draft, Goteborg, Sweden, 1980.

[BUC79] BUCHMANN, A. P. and A. G. Dale, "Evaluation Criteria for Logical Database Design Methodologies." Computer-Aided Design, 13:3, May 1979, 121-126.

[CAL67] CALESS, T. W. and D. B. Kirk, "Application of UDC to Machine Searching." Journal of Documentation, 23, 1967, 208-215.

[CLE77] CLEMONS, E., "An External Schema Facility to Support Database Update." Proceedings of the Joint US-USSR Seminar on Data Models and Database Systems. Austin, Texas, 1977.

[CLE79] CLEMONS, E., "Design of an External Schema Facility to Define and Process Recursive Structures." Proceedings of the Joint US-USSR Seminar on Data Models and Database Systems. Austin, Texas, 1979.

[COA73] COATES, E. J., "Progress in Documentation: Some Properties of Relationships in the Structure of Indexing Languages." Journal of Documentation, December 1973, 390-404.

[COA78] COATES, E. J., "Classification in Information Retrieval: The Twenty Years Following Dorking." Journal of Documentation, 34:4, December 1978, 288-299.

[COD79] CODD, E. F., "Extending the Database Relational Model to Capture More Meaning." ACM TODS 4:4, December 1979.

[COU73] COUGER, J. D., "Evolution of Business System Analysis Techniques." ACM Computing Surveys, 5:3, September 1973.

[CRE56] CRESTADORO, Andrea. The Art of Making Catalogues of Libraries. The Literary, Scientific and Artistic Reference Office, 1856.

[CUT76] CUTTER, Charles A. Rules for a Dictionary Catalog. 4th reprinting. Washington, D.C.: Government Printing Office, 1904.

[DEW76] DEWEY, Melvil. A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library. Amherst, Mass.: Amherst College, 1876.

[EDW59] EDWARDS, Edward. Memoirs of Libraries. Vol. 2. London, 1859.

[FAR73] FARRADANE, Jason. Relational Indexing. London, Ont.: School of Library and Information Science, University of Western Ontario, 1977.

[FRE68] FREEMAN, R. R. and Pauline Atherton. AUDACIOUS -- an Experiment with an On-line Interactive Reference Retrieval System Using the Universal Decimal Classification as the Index Language in the Field of Nuclear Science. AIP/UDC-7. New York: American Institute of Physics, 1968.

[HAM76] HAMILTON, M. and S. Zeldin, "Higher Order Software -- A Methodology for Defining Software." IEEE Trans. on Software Engineering, SE2:1, March 1976, 9-32.

[HAM78] HAMMER, M. and D. McLeod, "The Semantic Data Model: A Modelling Mechanism for Database Applications." Proc. 1978 SIGMOD Conf.

[HOP07] HOPWOOD, H. V., "Dewey Expanded." Library Association Record, 9, 1907, 307-322.

[HUF82] HUFF, S. L., "A Methodology for Supporting System Architects During Preliminary Design." Unpublished manuscript.

[HUL12] HULME, E. Wyndham, "Principles of Book Classification." Library Association Record, 14, 1912, 39-46, 174-181, 216-221. Reprinted as AAL Reprints no. 1. London: Association of Assistant Librarians, 1950.

[IBM61] IBM Corporation. Study Organization Plan. C20-8075. White Plains, N.Y., 1961.

[IBM75] IBM Corporation, Data Processing Division. HIPO - A Design Aid and Documentation Technique. GC-20-1851. White Plains, N.Y., 1975.

[INT57] INTERNATIONAL FEDERATION OF DOCUMENTATION. Proceedings of an International Conference on Classification for Information Retrieval. London: Aslib, 1957.

[KAH78] KAHN, B. K., "A Structured Logical Database Design Methodology." Proceedings, NYU Symposium on Database Design. May 1978.

[KAI11] KAISER, Julius O. Systematic Indexing. London: Vacher and Sons, 1911.

[KUH70] KUHN, Thomas S. The Nature of Scientific Revolutions. 2nd ed. Chicago: University of Chicago Press, 1970.

[LAN77] LANO, R. J. The N2 Chart. TRW Software Series, November 1977.

[LUH59] LUHN, H. P. Keyword in Context Indexing for Technical Literature. New York: International Business Machines, 1959.

[LUR77] LURCOTT, E. "F D , A System Management Tool." Data Systems Management Review, 1:4, August 1977.

[MIT80] MITTERMEIR, R. T. and R. T. Yeh, "Modelling Cc Structures using Semantic Nets." TR-. . College Park: University of Maryland, 1980.

[MYE73] MYERS, G. J. Composite Structured Design. New York: Van Nostrand Reinhold Co., 1978.

[NEE57] "The Need for a Faceted Classification as the Basis of all Methods of Information Retrieval." Appendix 2 in Proceedings of International Study Conference on Faceted Classification for Information Retrieval. London: Aslib, 1957.

[NEW57] NEWMAN, S. M., "Linguistics and Information Retrieval: Toward a Solution of the Patent Office Problem." Monograph Series in Linguistics and Language Studies, No. 10. Washington, D.C.: Georgetown University Press, 1957. pp. 103-111. [Reprinted in Journal of the Patent Office Society, 39:10, October 1957, 720-729]

[NEW58] NEWMAN, S. M., "Storage and Retrieval of Contents of Technical Literature." Office of Research and Development, Patent Office, 1958.

[NUN76] NUNAMAKER, J. F. and B. R. Kosynsky, "Computer-Aided Analysis and Design of Information Systems." CACM, 19:12, December 1976, 676-687.

[PETE77] PETERSON, J. L., "Petri Nets." ACM Computing Surveys, 9:3, September 1977, 223-252.

[PETR75] PETRI, C. A., "Interpretations of Net Theory." Institut fur Informations Systemforschung Interner Berecht 15-07, July 1975.

[RAN33] RANGANATHAN, S. R. Colon Classification. Madras: Madras Library Association, 1933. [6th edition published by Asia Publishing House, 1960]

[RAN58] RANGANATHAN, S. R. Prolegomena to Library Classification. 2nd ed. London: Association of Assistant Librarians, 1958.

[ROS77] ROSS, D. T. and E. E. Schoman, Jr., "Structured Analysis for Requirements Definition." IEEE Trans. on Software Engineering, SE3:1, January 1977.

[ROS68] ROSTRON, R. M., "The Construction of a Thesaurus." Aslib Proceedings, 20:3, March 1968, 181-187.

[ROU79] ROUSSOPOULOS, N., "CSDL: A Conceptual Schema Definition Language for the Design of Data Base Applications." IEEE Trans. on Software Engineering, SE5:5, September 1979.

[SAL76] SALTER, K., "A Methodology for Decomposing System Requirements into Data Processing Requirements." Proceedings of the Second International Conference on Software Engineering, October 1976, 91-98.

[SAY18] SAYERS, W. D. Berwick. An Introduction to Library Classification. London: Grafton and Co., 1918.

[SCH78] SCHKOLNICK, M., "A Survey of Physical Database Design Methodology and Techniques." Proceedings of the Fourth International Conference on Very Large Data Bases. Berlin, 1978.

[SMI78] SMITH, J. M. and D. C. P. Smith, "Principles of Database Conceptual Design." Proceedings NYU Symposium on Database Design. May 1978.

[SOF76] SOFTECH, SADT, the Softech Approach to System Development. The Software Technology Company, January 1976.

[TAG77] TAGGART, W. M., Jr. and M. O. Tharp, "A Survey of Information Requirements Analysis Techniques." ACM Computing Surveys, 9:4, December 1977, 273-290.

[TEI74] TEICHROEW, D., "Improvements in the System Life Cycle." Information Processing, 76, 1976, 972-978.

[TEI77] TEICHROEW, D. and E. A. Hershey, III, "PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems." IEEE Trans. on Software Engineering, SE3:1, January 1977, 41-48.

[TEO82] TEOZEY, T. J. and J. P. Fry. Design of Database Structures. Englewood Cliffs: Prentice-Hall, 1982.

[UDC33] UNIVERSAL DECIMAL CLASSIFICATION. English edition. London: British Standards Institution, 1933-.

[VIC59] VICKERY, B. C. Classification and Indexing in Science. 2nd ed. London: Butterworth, 1959.

[WIL79] WILSON, M. L. A Semantics-Based Requirements and Design Method. TR-03.072. San Jose: IBM, 1979.

[YAO78] YAO, S. B., S. B. Navathe, and J. L. Weldon, "An Integrated Approach to Logical Database Design." Proceedings NYU Symposium on Database Design, May 1978.

APPENDIX A

DEVELOPMENT OF A DATA ELEMENT DICTIONARY AND LOCATOR SYSTEM

Some idea of what an agency level dictionary and locator system might be like, and of what its construction might involve, can be illustrated by an examination of the experiences of the Department of Energy and its predecessors and constituent units.

In 1978 Dr. Lincoln Moses, then the Administrator of the Energy Information Administration (EIA), set up a task force, chaired by Bruce Dwyer of EIA's Office of Planning and Evaluation, to report on past and current efforts to manage EIA's energy information, particularly the Information Element Dictionary (IED), and to propose a plan for the future. This report (the Dwyer Report) was issued in November 1978.

The Dwyer Report reviewed a number of efforts:

- Federal Energy Information Locator System (FEILS), 1976;
- Information Element Dictionary (IED), 1978;
- Information Access Corporation's (IAC) index to oil and gas reserve information, then in process, completed 1979;
- Federal Energy Data Index (FEDEX), then in a pilot stage;
- Logistics Management Institute (LMI) NEIS conceptual design, and sample vocabulary and grammar, 1978.

There is no need to present or discuss the fine detail of the Dwyer Report's comments on these systems or projects, but some account is needed because of the impact of the Report's recommendations on the later development of the EIA faceted classification and the EIA Data Resources Directory (DRD).

The FEILS was intended only for location at the system and database level, though it had done yeoman service in analyzing the very complex world of energy information in 1976. The IAC work was intended to be aimed at the data element level, but actually indexed at the table level. FEDEX described publications and the tables within them only down to the table level. LMI addressed broad organizational responsibilities. Only the IED addressed the data element level, i.e., blocks or blanks on forms, intersections of row and column in tables, etc.

There were similarities in the approaches of most systems: for example, in the recognition of distinct aspects of energy information, like ENERGY SOURCE, ENERGY FUNCTION, UNITS OF

MEASUREMENT. Not all systems recognized all possible aspects (LMI did not even include SPACE and TIME) and there was little vocabulary control. FEILS had developed a limited list of terms that was used in a different categorization by the IED, but it was too broad for use at the data element level and had to be extended by the use of less controlled terms and completely uncontrolled qualifiers.

Most systems fell into disuse because they were too limited or too complex to use. The IED described each data element on a separate (printed) page and as a result was 29 volumes in length - too cumbersome for easy use. The IAC work recognized the component vocabulary types of MATERIAL, FUNCTION, UNITS, SPACE and TIME, but the vocabulary basis for MATERIAL and FUNCTION was taken from FEILS, and was thus not organized in mutually exclusive and collectively exhaustive clusters. (A later adverse comment, not in the Dwyer Report, was that the IAC indexing took further vocabulary and all hierarchical structuring without question or modification directly from the tables.)

In the light of these comments, the Dwyer Report called for a new approach to an improved IED that would be able to:

- o determine the availability and location of energy data;
- o provide descriptive information about the data and the places they are located;
- o identify similar or like data;
- o discover gaps in the availability of data;
- o show the relationships between the components of the information processing network.

The new IED would be based on a highly structured classification scheme that would include facets of ENERGY SOURCE, FUNCTION, MEASUREMENT, SPACE, TIME, PROPERTIES OF ENERGY SOURCES, OTHER MATERIALS and OTHER FUNCTIONS. It would cover all EIA data instruments and it would address the most specific data element level, rather than any higher, aggregated level. Lastly, the system should be automated and interactive.

In other words, two parallel tasks were necessary: to develop the highly structured classification scheme (since none existed); and to develop interactive software that would allow the system to accept and store very complex statements that were the data element descriptions, and also allow the user to interrogate the system under guidance, to identify and locate any complex data element description by all or any of its component data elements. For example, the classification would have to be able to include (i.e., to name) a complex data element as specific as:

THOUSANDS OF SHORT TONS OF BITUMINOUS COAL SHIPPED MONTHLY BY BARGE FROM A GIVEN MINE TO AN ELECTRIC UTILITY

and to allow it to be retrieved by any search, such as:

"What data elements include THOUSANDS OF SHORT TONS OF BITUMINOUS COAL SHIPPED MONTHLY TO ELECTRIC UTILITIES"

or:

"THOUSANDS OF SHORT TONS OF BITUMINOUS COAL [moved] MONTHLY"

or even:

"all references to ELECTRIC UTILITIES".

Further, if an enquiry received an unsatisfactory response, the system should be able to recommend alternative search terms to the user - more general, more specific or related terms.

The EIA Office of Energy Information Validation (OEIV) was given responsibility to produce an RFP for a new IED. Later, responsibility for the RFP and the overall direction of the development of the new IED, the Data Resources Directory (DRD), was transferred to the newly established National Energy Information System (NEIS). NEIS called on the services of two experts: in faceted classification and in database management, respectively Professor C. D. Batty and Dr. Edgar H. Sibley, both of the University of Maryland, and both now with the Alpha Omega Group, Inc.

THE DEVELOPMENT OF THE CLASSIFICATION SCHEME FOR THE DICTIONARY

There were two immediate tasks. The first was to demonstrate the feasibility of using faceted classification principles in the development of the DRD; the second was to produce the RFP, and, if faceted classification did prove to be an appropriate approach, to incorporate sufficient information in the RFP to make it clear to offerors what abilities they should demonstrate.

Further, also assuming that faceted classification did prove to be the appropriate approach, two more tasks became apparent. The first of these was to investigate a number of questions concerning EIA's use of faceted classification, and to produce what became known as "clarification papers" for the guidance of EIA personnel and a future contractor. The second was to develop a prototype faceted classification scheme that would be Government Furnished Information (GFI) for the new DRD, and thus help to ensure that the contractor would follow EIA policy.

At this point it should be stated that the general structure of such a faceted classification is as follows.

The whole field is divided into facets, each of which reflects only one characteristic, or family of characteristics. A simple facet, e.g., ENERGY SOURCES, will then be arranged as a simple hierarchy or tree of terms and in that hierarchy any set of

coordinate terms (siblings) is called an array. For example, in ENERGY SOURCES, one array contains COAL, PETROLEUM, NATURAL GAS, etc.; another (within COAL) contains ANTHRACITE, BITUMINOUS COAL, LIGNITE, etc.

A complex facet, however, may have several related characteristics, e.g., in FUNCTIONS/ACTIVITIES the characteristics of the life cycle, of movement, of financial dealing. These characteristics are used to develop subfacets, which, if simple, are then arranged as hierarchies or trees. It is of course possible to go to a third level of the application of characteristics. Typically subfacets are mutually exclusive (they will contain different concepts) and are combinable, e.g., the COST of STORING for PROCESSING. It is the function of a faceted classification structure to ensure that the knowledge base is analyzed into mutually exclusive, simple trees. As can readily be imagined a highly complex knowledge base could not be listed as a two-dimensional taxonomy; the faceted structure offers clusters and lists of easily identifiable components, with which to assemble complex data element descriptions.

The examination of the appropriateness of faceted classification for EIA data was carried out by analyzing a number of sources of EIA information, initially and principally the Annual Report to Congress and its index, an issue of the Monthly Energy Review, and a small selection of forms covering coal, petroleum, and natural gas. This analysis provided a preliminary list of facets and some subfacets:

ENERGY SOURCE
  Natural sources
  Derived sources

ENERGY PROPERTIES

ENERGY FUNCTIONS
  Production
  Consumption

AGENTS or TOOLS

SPACE

TIME

PARTICIPANTS
  Active (suppliers)
  Passive (consumers)

UNITS OF MEASUREMENT

OTHER MATERIALS

OTHER FUNCTIONS

The list was further developed as an outline classification scheme, and fitted with a two-digit notation. It was then used to analyze and describe data elements from more EIA forms.

It became clear that faceted classification did offer a useful approach. The demonstration revealed a number of points:

- o facet analysis could handle the variety of aspects or components of energy data and data descriptions;
- o once the facets were established, analysis of even complex data elements could be done
  - more quickly
  - more consistently
- o there were more facets than had been supposed by the Dwyer Report, and that therefore the classification approach was even more appropriate;
- o the development of a scheme on faceted lines would be simpler because each facet could be developed as a fairly simple taxonomy;
- o the resulting scheme would be easier for indexers to use because they would
  - analyze the data elements by facets
  - scan simple vocabulary taxonomies within facets

The preliminary classification is shown in Table A:1. Of course, in order to arrive at this two-level outline scheme a number of facets had to be developed in greater detail, in some cases to a five-level scheme.

The original facets of OTHER MATERIALS and OTHER FUNCTIONS were discarded and their potential content was merged with other facets. For example, much of OTHER MATERIALS (i.e., non-energy) proved to be AGENT/INSTRUMENT, and much of OTHER FUNCTIONS proved to be acceptable to the FUNCTION facet. Some OTHER FUNCTIONS remained, and was later incorporated into an EXTERNALITIES facet that included topics like ECONOMICS, ENVIRONMENTAL CONCERNS, etc.

Subsequent development of the scheme used four sources:

- o a sample collection of forms on coal and petroleum (EIA 6, EIA 7, EIA 9, EIA 25, EIA 50, EIA 52, EIA 79);
- o publications and tables (Annual Report to Congress, the Monthly Energy Review);

- other classifications and listings of energy data (e.g., the EDB Thesaurus, the National Energy Accounting System);

- discussion with EIA personnel and consultants.

A number of changes were made in this phase of the development of the classification.

The organization of the ENERGY SOURCE facet was changed from NATURAL and DERIVED to one based on the energy sources themselves, subdivided into natural and derived classifications. Thus, in the first version 01 COAL (in NATURAL SOURCES) was separated in the classification from 02 12 COKE AND BREEZE (in DERIVED SOURCES) by almost two pages of intervening NATURAL SOURCES, e.g., 01 2 NATURAL GAS, 01 3 PETROLEUM (= crude oil). Furthermore, 01 3 PETROLEUM was separated from 02 242 GASOLINE by a page and a half. In the revised version, 01 COAL AND COAL PRODUCTS contains 01 1 COAL and 01 2 COKE AND BREEZE and 02 PETROLEUM AND PETROLEUM PRODUCTS contains 02 1 PETROLEUM (= crude oil) and 02 42 GASOLINE. Table A:2 shows this part of the schedules in more detail.

As more detailed work was done with the classification, a number of problems were perceived in the organization of PETROLEUM PRODUCTS: DIESEL, NO.2 DISTILLATE, and NO.2 HOME HEATING OIL are effectively the same, but are grouped and listed differently depending on a number of external factors. At first the main organization of this part of the ENERGY SOURCES facet followed the chemical constituent order reflecting the fractionating column, and terms like DIESEL and HOME HEATING OIL were fitted into the nearest appropriate place. Such a practice suited neither the chemists and engineers, who objected on purely chemical grounds, nor the economists and statisticians, who claimed that for their purposes a fuel was what its container or bill of lading announced it to be. Finally the PETROLEUM PRODUCTS section of the scheme was broken into two parallel subsections (or subfacets): PRODUCTS BY CHEMICAL CONSTITUENT and PRODUCTS BY USE. Instructions were included when one or another index term should be used, e.g., MOTOR GASOLINE appears at a particular point in the fractionating column, but it, and its varieties, like REGULAR, PREMIUM, etc., were all listed fully in the "BY USE" subfacet, with an instruction in the scheme to use that list. Relevant sections of the classification are shown in Table A:3 by way of illustration.

The above example has been included to illustrate the kind of settling down process vital to the development of a vocabulary and a classification structure that must keep different classes of users happy.

Another, and potentially more serious example occurred in the FUNCTIONS/ACTIVITIES facet. The early versions of the scheme had included a simple list of the functions involved:

DISCOVERING
CONSTRUCTION
EXTRACTING
PROCESSING
CONVERTING
CONSUMING
RECYCLING
MOVING
STORING

This was clearly not sufficiently organized, and so the first five terms were subsumed under PRODUCTION; CONSUMING was clearly CONSUMPTION, but the remaining terms were not as easily handled. MOVEMENT and STORAGE can take place at any point in the energy life cycle.

The first outline scheme (shown in Table A:1) offered a compromise solution that located them between PRODUCTION and CONSUMPTION. That outline also added the concept of ECONOMIC MOVEMENT (e.g., sales) to PHYSICAL MOVEMENT. A much more serious and fundamental problem then arose because the arrangement naturally invited the suggestion that the subfacets and their constituent arrays (the siblings) should balance, to give a statistically satisfactory picture of the energy life cycle in terms of amounts, dollar values, etc.

Development in this direction introduced inordinate complication. For example, to include EXPLORATION and DISCOVERY as part of an economically balanced picture would have meant the introduction of highly complex economic considerations like investment that appear only rarely in energy forms and reports and even then not in direct connection with the energy life cycle. In fact economic considerations had already been introduced into the scheme in a special facet including concepts like the GNP that might affect several other facets.

In the end, the FUNCTIONS/ACTIVITIES facet was revised to show again the energy life cycle as its principal subfacet, with TRANSPORTATION, STOCKS and STORAGE as a second subfacet, FINANCIAL ACTIVITY as a third subfacet (since its terms can apply to TRANSPORTATION) and a fourth subfacet of general modifiers.

Only one other major change was made to the scheme: the bringing forward of PARTICIPANTS to a position immediately before SPACE. This was not a difficult change to make. Facets are mutually exclusive and independent; therefore the change involved only editing the classification file to move the facet, making a global change to the first character of notation, and printing out a new version.

The kinds of problem discussed above are typical of the development of such a vocabulary and classification. Also typical was the small degree to which the general structure of the classification changed during the course of its development. The outline proposed after the first few weeks of analysis is virtually

unchanged today. Such a feature of a structured vocabulary is significant in the development of a dictionary for a locator system.

Another significant point deserves to be made. The use of a faceted structure dramatically reduces the physical size of the vocabulary. The EIA classification is able to handle the most specific and complex data elements in the EIA data instruments, yet it is only some 50 pages long. The corresponding vocabulary in use in the Department of Energy's Energy Data Base (the EDB Thesaurus) is several hundred pages long.

TABLE A:1

PRELIMINARY ENERGY DATA CLASSIFICATION

0 ENERGY SOURCE

01 Natural sources (eg, Coal, Petroleum, Organic)

02 Derived sources (eg, Coal products, Petroleum products, Electricity)

1 PROPERTIES OF ENERGY SOURCES

11 Content
12 Physical
13 Chemical
14 Other

2 ENERGY FUNCTIONS

21 Supply
22 Movement (including economic movement, ie, buying and selling)
24 Demand

3 AGENT/INSTRUMENT

31 Manual equipment
35 Earth moving equipment
36 Transport equipment

4 SPACE

42 Politico-geographical (eg, Maryland)
43 Administrative (eg, BOM districts)
44 Geological (eg, Peruvian Basin)
45 Spheres of enterprise (eg, OPEC)
46 Geographical (eg, off-shore)
47 Orientation (eg, north)
49 Direction (eg, to a place)

5 TIME

51 Chronological
52 Duration
53 Frequency
56 Other aspects (eg, preceding)

6 PARTICIPANTS

61 Active (eg, suppliers)
62 Channel (eg, trucking companies)
63 Passive (eg, consumers)

7 UNITS OF MEASURE

71 Monetary
72 Dimension
73 Area
74 Volume
75 Quantity
76 Density
77 Heat/energy exchange
78 Proportion

TABLE A:2

PROTOTYPE ENERGY DATA CLASSIFICATION

0	ENERGY SOURCE
	NB: Use 0 ENERGY SOURCE in any data element description that refers generally to energy sources, e.g., MILLIONS OF BTUs OF ALL ENERGY USE
01	COAL AND COAL PRODUCTS
01 1	COAL (Natural Source)
01 12	Anthracite
01 122	Meta-anthracite
01 123	Anthracite
01 124	Semi-anthracite
01 13	Bituminous
01 132	Low volatile bituminous
01 133	Medium volatile bituminous
01 134	High volatile bituminous
01 1342	High volatile A bituminous
01 1343	High volatile B bituminous
01 1344	High volatile C bituminous
01 14	Sub-bituminous
01 142	Sub-bituminous A
01 143	Sub-bituminous B
01 144	Sub-bituminous C
01 16	Lignite
01 162	Lignite A
01 163	Lignite B
01 17	Peat
01 18	Methane
01 19	Mines
01 192	Coal bed
01 1922	Recoverable coal
01 1923	Refuse
01 193	Working mire
01 194	Idle
01 195	Abandoned
01 196	Out of business

01 2-9 COAL PRODUCTS

01 2	Coke and breeze
01 22	Coke
01 23	Breeze
01 24	Coke etc produced in beehive
01 25	Coke etc produced in oven
01 4	Coal tar
01 5	Pitch
01 6	Fuel briquettes
01 7	Ammonia products
01 9	Other coal derivatives

02	PETROLEUM AND PETROLEUM PRODUCTS
02 1	Petroleum (natural source); crude oil
02 11	Natural petroleum
02 16	Natural gas (wet)
02 17	Other naturally occurring sources
02 172	Tar sands
02 173	Oil shale
02 175	Natural gasoline
02 176	Other natural gas liquids

02 2-9 PETROLEUM PRODUCTS

02 2	Natural gas (dry)
02 22	Methane
02 3	LPG and olefins
02 32	Ethylene and ethane
02 323	Ethelene
02 324	Ethane
02 33	Propylene and propane
02 332	Propylene
02 333	Propane
02 34	Butylene and butane
02 342	Butylene
02 343	Butane
02 3432	Iso-butane
02 344	Iso-pentane

02 4 Gasoline and naphthas

02 42 Gasoline
02 422 Motor gasoline
02 4222 Unleaded motor gasoline
02 42222 Regular unleaded motor gasoline
02 42223 Premium unleaded motor gasoline
02 4223 Leaded motor gasoline
02 42232 Regular leaded motor gasoline
02 42233 Premium leaded motor gasoline
02 423 Aviation gasoline

02 43 Naphthas
02 433 Naphtha type jet fuel
02 434 Naphtha 400
02 435 Special naphthas
02 4352 Naphthas less than 400
02 4353 Naphthas greater than 400
02 436 Iso-pentane

02 5 Fuel oils

02 52 Kerosene

NB: Details are included because data elements refer to KEROSENE by name. It is noted that RANGE OILS is the same as no.1 DISTILLATE used in a different context.

02 522 Range oils
02 523 Jet fuels

02 53 Distillate fuel oils
02 531 no.1 distillate fuel oil
02 532 no.2 distillate fuel oil
02 534 no.3 distillate fuel oil

02 54 Residual fuel oils
02 545 no.5 residual fuel oil
02 546 no.6 residual fuel oil
02 547 Bunker C
02 548 Navy special

TABLE A:3

DEVELOPED ENERGY DATA CLASSIFICATION

A	ENERGY SOURCES
Af	ENERGY SOURCES BY TYPE
Af b	PETROLEUM AND PETROLEUM PRODUCTS
Af Bf	PETROLEUM
Af bfe	Crude oil
Af bfef	Domestic crude oil
	Note: Do not use "Domestic" (Bue) from the SOURCE QUALIFIER facet with "Crude oil".
Af bfefk	Alaskan North Slope crude oil
Af bfer	Foreign crude oil
	Note: Do not use "Foreign" (Buk) from the SOURCE QUALIFIER facet with "Crude oil". For specific types of foreign crude oil, see Appendix A.
Af bfk	Tar sands
Af bfs	Oil shale
Af br	PETROLEUM PRODUCTS
Af bre	Petroleum products (light to heavy)
Af breb	Paraffins and olefins
Af brebe	Ethane/ethylene
	Ethane (Afcrfsc)
Af brebk	Propane/propylene
	Propane (Afcrfske)
	Ethane-propane mixtures (Afcrfsg)
Af brebs	Butane/butylene
	Butane (Afcrfskk)
	Butane-propane mixtures (Afcrfsks)
	Pentane (Afcrfsp)
Af bred	Gasoline and naphthas
Af bredf	Gasoline
	Motor gasoline (Afbrkrce)
	Aviation gasoline (Afbrkrff)
Af bredr	Naphthas
Af bredre	Naphthas less than 400
Af bredrk	Naphtha 400
Af bredrs	Naphthas greater than 400
	Special naphthas (Afbrsv)

Af bref	Fuel oils
Af brefe	Kerosine
Af brefk	Distillate fuel oils
Af brefke	No. 1 distillate fuel oil
	No. 1 heating oil (Afbrkrne)
	No. 1-D diesel fuel (Afbrkrcke)
Af brefkk	No. 2 distillate fuel oil
	No. 2 heating oil (Afbrkrnk)
	No. 2-D diesel fuel (Afbrkrckk)
Af brefks	No. 4 distillate fuel oil
	No. 4 heating oil (afbrkrns)
	No. 4-D diesel fuel (Afbrkrcks)
Af brefs	Residual fuel oils
Af brefsf	No. 5 residual fuel oil
Af brefsr	No. 6 residual fuel oil
Af brefsrf	Bunker C
Af brefsrr	Navy special fuel oil
Af breh	Technical oils
Af brek	Waxes
Af brekf	Microcrystalline waxes
Af brekr	Paraffin waxes
Af brekrk	Crystalline waxes
Af brekrkk	Fully refined crystalline waxes
Af brep	Lubricating oils and greases
Af brepe	Lubricating oil basestocks
Af brepef	Bright stock
Af breper	Neutral lubricating oil basestock
Af brepk	Lubricating oils
Af breps	Lubricating greases
Af brer	Road oils
Af bret	Asphalt
Af brev	Petroleum coke
Af brevf	Marketable petroleum coke
Af brevfk	Green petroleum coke
Af brevr	Catalyst petroleum coke
Af brk	Petroleum products by use
Af brkf	Petroleum products by use in product preparation
Af brkfc	Refinery feedstocks
Af brkfcf	Fresh feed
Af brkfcr	Recycled feed
Af brkfg	Petrochemical feedstocks
Af brkfk	Petrochemical blendstocks
Af brkfp	Gasoline blending components
Af brkft	Absorption oil

Af	brkr	Petroleum products by energy use
Af	brkrc	Automotive fuels
Af	brkrce	Motor gasoline
Af	brkrcef	Unleaded motor gasoline
Af	brkrceff	Regular unleaded motor gasoline
Af	brkrcefr	Premium unleaded motor gasoline
Af	brkrff	Aviation gasoline
Af	brkrfr	Jet fuels
Af	brkrfrf	Naphtha-type jet fuel
Af	brkrfrr	Kerosine-type jet fuel
Af	brkri	Range oils
Af	brkrn	Heating oil
Af	brkrne	No. 1 heating oil
Af	brkrnk	No. 2 heating oil
Af	brkrns	No. 4 heating oil
Af	brkrr	Synthetic crude oil (oil shale)
Af	brkru	Synthetic natural gas (petroleum)
Af	brs	Other petroleum product groupings
Af	brsb	Unfractionated stream
Af	brse	Still gas
Af	brsh	Liquefied refinery gases
Af	brsk	Topped crude
Af	brsp	Oils
Af	brspk	Gas oils
Af	brss	Refined aromatics
		Benzene (Afekhre)
		Toluene (Afekhrk)
Af	brsv	Special naphthas
Af	c	NATURAL GAS AND NATURAL GAS PRODUCTS
Af	cf	NATURAL GAS
		Note: For wet natural gas and dry natural gas use "Natural gas" and "Wet" (Bcqf) or "Natural gas" and "Dry" (Bcqr).
Af	cfk	Liquefied natural gas
		Note: For Synthetic natural gas from all sources, gas made from natural gas liquids use "Synthetic natural gas (natural gas liquids)" (Afcrrt).

Af cr	NATURAL GAS PRODUCTS
Af crf	Natural gas products (light to heavy)
Af crfe	Hydrogen
Af crfk	Methane
Af crfkk	Methane (natural gas)
	Methane (coal) (Afekr)
	Methane (organic sources) (Afgrd)
Af crfs	Natural gas liquids
Af crfsc	Ethane
Af crfsg	Ethane-propane mixtures
Af crfsk	Liquefied petroleum gases
Af crfske	Propane
Af crfskk	Butane
Af crfskkf	Normal butane
Af crfskkr	Isobutane
Af crfsks	Butane-propane mixtures
Af crfsp	Pentane
Af crfspk	Isopentane
Af crfst	Natural gas liquids heavier than pentane
Af crr	Other natural gas product groupings
Af crrd	Liquefied gases
Af crrdk	Liquefied plant gases
	Liquefied petroleum gases (Afcrfsk)

APPENDIX B

GLOSSARY

Array

The list of coordinate terms in a classification scheme at any level of division in a facet or subfacet.

Attribute

A data item containing information about an entity, e.g., characteristic property.

Basic data element (BDE)

A single component or data item in a complex data element, e.g., GEARBOX, NUMBER, and TESTING are each basic data elements within the complex data element NUMBER OF GEARBOXES TESTED. In the developmental phase of the classification, when a basic data element is derived from an examination of a sample document, it is called a raw BDE. Once a raw BDE has been accepted into the classification, and its authoritative name established, it is called a refined BDE.

Chain

The series of steps of subdivision in a classification scheme leading from a general point in the tree to a specific point.

Complex data element (CDE)

A collection of basic data elements that together describe an event or phenomena in the information environment. In the developmental phase of the classification, when complex data elements are encountered in the sample documents, etc., they are called raw CDE's, and are decomposed into raw BDE's. When a complex data element description is made up as part of the picture of the information environment (i.e., an assembly of refined BDE's from the classification), it is called a refined CDE.

Conceptual Schema

The overall logical structure of a database creating a conceptual model of data in an application environment (e.g., enterprise).

Containers

Forms, reports, machine-readable files, etc., or distinct parts of them, in which complex data elements (sometimes records) appear or are stored, and from which basic data elements are derived in order to develop the classification.

Data description language

A language for describing data and a database schema in accordance with some model/approach (e.g., relational, network, hierarchical) and level of description (e.g., logical, physical).

Data item

The smallest named unit of data that has meaning in describing information (synonymous with basic data element or field).

Data manipulation language

A language used to manipulate data stored in a database and to transfer it to/from a user program.

Database

A collection of interrelated data stored together to serve one or more applications and controlled by a database management system.

Database management system

Software required for describing (predefining), creating, and using/manipulating a database.

Domain

The collection of data items (fields) of the same type in a relation.

Entity

In the context of database design an entity is anything about which data is recorded. In the context of classification an entity is a passive substance, person,

system, or mental construct (mentefact) that is the subject of a process, operation, problem, etc. The context has been made clear on each use of the term in this Report.

External schema

The application view of a database (subschema).

Facet

A cluster in the classification scheme of basic data elements sharing a single characteristic. Like all members of an array at any level of the classification, they must be mutually exclusive and collectively exhaustive.

Facet type

Functionally equivalent to a characteristic. The facet type is the name of a facet that is not yet populated.

File

A set of similarly constructed records.

Focus (pl. Foci)

A specific term or node in a tree of classification; a focus may be a class of things susceptible to further subdivision.

Fundamental category

The highest level of abstraction in a generalized classification structure.

General category

The second level of abstraction in a generalized classification structure. The list of general categories acts as a suprastructure or overall checklist in the initial analysis of an information environment, and may also be used later, in a menu-driven retrieval system for a user checking the information environment.

Information environment

The information about an enterprise and its activities, together with the mechanisms and instruments (like forms, documentation, screen layouts, files, etc.) for collecting,

manipulating, transmitting, and reporting that information.

Internal schema

The physical structure of a database defined for the efficient storing of data.

Levels

Sub-trees sharing the same general characteristic whose foci may modify foci from other sub-trees. Typically, levels contain respectively whole things, organs or parts of things, parts of organs or of parts, etc.

Record

An entity considered as a group of related data items and treated as a unit by an application program; cf Complex data element.

Relation

A two-dimensional array of data items (a flat file).

Relationship

A link between entities.

Rounds

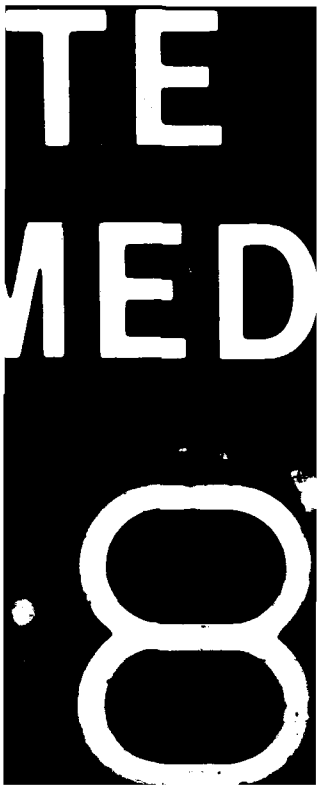
The recurrence of a fundamental or general category at an unusual place in the ordered list of facet types. Rounds occur most frequently when THINGS are assigned to the role of TOOLS involved in a PROCESS.

Subfacet

A facet-like cluster within a facet in the classification scheme. There must be at least two subfacets for one to exist at all. Subfacets occur when the characteristic that gives rise to a facet may be further decomposed into mutually exclusive and combinable subcharacteristics.

Subfacet type

The name of an unpopulated subfacet that has been recognized in a facet in the classification scheme.


TE
MED
8